KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Relational Visual Recognition

Laura Antanas

Supervisor:
Prof. Dr. Luc De Raedt
Co-supervisor:
Prof. Dr. ir. Tinne Tuytelaars

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

June 2014

Relational Visual Recognition

Laura ANTANAS

Examination committee:
Prof. Dr. Adhemar Bultheel, chair
Prof. Dr. Luc De Raedt, supervisor
Prof. Dr. ir. Tinne Tuytelaars, co-supervisor
Prof. Dr. ir. Herman Bruyninckx
Prof. Dr. ir. Maurice Bruynooghe

Dr. ir. Kurt Driessens
(Maastricht University, The Netherlands)
Prof. Dr. Paolo Frasconi
(Università degli Studi di Firenze, Italy)
Prof. Dr. David Hogg
(University of Leeds, UK)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

June 2014

Uitgegeven in eigen beheer, Laura Antanas, Celestijnenlaan 200A 3001 Heverlee, B-3001 Heverlee (Belgium)

ISBN 978-94-6018-855-8

D/2014/7515/79

Abstract

In contrast to statistical visual recognition, relational visual recognition aims at employing relational representations for solving visual recognition problems. For high-level tasks involving complex objects and scenes, low- and mid-level visual features do not always suffice. In these cases it is the component objects, their structure and semantic configuration that guides recognition. They are best described in terms of relational languages or (higher-order) graphs. Relational approaches enjoyed popularity in the early vision work. Convenient at that time given the limitations of the hardware, data, scientific technologies and low-level vision routines, relational representations are rarely used in visual recognition today. This is mainly due to their pure symbolic nature. Nevertheless, recent successes in combining them with statistical learning principles and the maturity of the aforementioned resources motivates us to reinvestigate their use. Starting from low-and mid-level solutions and building on top of them, (statistical) relational learning gives the perspective of moving towards more general, complete and effective relational visual recognition systems.

The thesis makes several contributions in this direction, three in the field of computer vision and two in the field of robotics. We first introduce a new relational distance-based framework for hierarchical image understanding. Applied to the house facade domain, the relational distance shows good detection results, while demonstrating the interplay between structural and appearance-based aspects. The second contribution is the use of a kernel-based relational language for scene classification and tagging. Part of this contribution is the employment of the kernel-based language to understand images of houses. These recognition tasks use a similar relational representation and language, showing its generality and benefits. Our third contribution is a probabilistic logic pipeline for task-dependent robot grasping. It contains a new module based on causal probabilistic logic and symbolic object parts, such that, given a set of probabilistic observations about the world, it can semantically reason about object category, suitable tasks and pre-grasp configurations with respect to the intended task. Experimental results, including those obtained with a real robot

platform, confirm the importance of high-level reasoning and world-knowledge for robot grasping, as opposed to using solely local object shape information. Further, in the context of robot grasping, our fourth contribution is a relational approach to numerical feature pooling. It combines numerical shape features, qualitative spatial relations and kernels for graphs to recognize graspable object points. Finally, we contribute with the use of sequential statistical relational techniques to capture underlying concepts in video streams. In particular, we focus on monitoring card games and learning to detect fraudulent sequences.

Overall, the experimental results provide evidence that *we can develop effective and real-world relational visual recognition systems that benefit from statistical relational learning.*

Acknowledgments

Working on this Ph.D. thesis has been a challenging, but also an exciting and rewarding experience. Now, at the end, I would like to thank several people. Without their support, advise and help I never would have been able to get this far.

First and mostly, I would like to greatly thank my supervisors Luc De Raedt and Tinne Tuytelaars. They always encouraged me to look for ideas and have inspired me with great pointers and suggestions. They always took the time to discuss my ideas and problems, write many and useful comments on my drafts. I am especially thankful for the support they gave me in the most difficult times, when I could not immediately see the results of my work. Their persistent belief in me was a great motivation during this Ph.D.

Next, I want to kindly thank all the members of my jury, Maurice Bruynooghe, Herman Bruyninckx, Kurt Driessens, Paolo Frasconi and David Hogg for carefully reading my text and for their highly appreciated feedback. Their suggestions improved the content of this dissertation. I also want to thank Adhemar Bultheel for chairing the defense.

Research involves working with many people. The help and input I received during my Ph.D. was of great value. My thanks go towards the entire DTAI research group for creating a motivational and enjoyable environment. I would also like to thank the people with whom I wrote papers and discussed research ideas, who helped me with feedback or made my time at work more fun by sharing an office, going for coffee or having nice chats. I start with Kurt D, Tom and Robby, my first officemates in the DTAI group. Tom and Kurt D, thank you for your help on our first paper together. It opened my appetite for research and gave me a great and highly motivational start in my Ph.D. Next, I want to thank Angelika, Bernd, Bogdan, Davide, Fabian, Guy, Ingo, Jose, Mathias, McElory, Tias, Thanh for not only being wonderful colleagues or officemates, but also my friends, helping me in difficult moments both on the professional

and personal side. I also thank Francesco, Martijn and Parisa for the times we shared together as officemates. Furthermore, I want to thank my co-authors: Bogdan, David, Fabian, Fabrizio, Jose, Kristian, Marion, Martijn, McElory, Paolo, Plinio, Niels, Thomas, Wannes. Special thanks to Mathias, McElory, Ingo, Fabrizio, Paolo, Kurt DG and Francesco for our long discussions on different research ideas. I also want to thank Albrecht, Anton, Bjorn, Dimitar, Fernando, Hendrik, Jan Ramon, Jesse, Joaquin, Joris, Kostas, Siegfried, Theo and Wannes for their help on different work-related issues during these years.

Furthermore, I would like to thank my friends. They have always encouraged me to finish my work and always offered me moral support. It is an honour to have you as friends.

Finally, I sincerely thank my family. First, I want to thank my husband, Cip, for his patience and understanding during these years. There were days when I had to work very late or I could not spend more time with my family because a conference deadline is a deadline! I deeply thank my parents for supporting and helping me throughout all the study years, for creating an environment that allowed me to focus on my studies, for always encouraging me to do what I was passionate about and follow my dreams. I want to thank my sister Simona for being such a wonderful friend. The very special thanks go to my son, David, for being such a good child. Thank you so much for your patience my sweet baby. David, baietelul meu drag, mama iti multumeste pentru rabdare. David si Ana, mama va iubeste foarte mult. Va multumesc draga mea familie din suflet!

Contents

List of Figures

List of Tables

Overture

Chapter 1

Introduction

1.1 Context

1.1.1 Machine Learning, Computer Vision and Robotics

Artificial Intelligence (AI) has the long-standing goal of building intelligent machines that can perceive, think and act in similar ways as humans [Landwehr, 2009]. This definition makes AI a big challenge to achieve, but also an inspiration to many researchers. Driven by this goal, AI today is an important field of computer science that can successfully solve tasks in many real-world applications, such as natural language understanding, drug discovery, fraud detection, 3D reconstruction or autonomous robot navigation.

This dissertation is situated at the intersection of three subfields of AI: *machine learning*, *robotics* and *computer vision*.

Machine learning is concerned with building systems that improve their performance on a task with experience, beyond human experts. Machine learning systems typically *learn* concepts from examples, either by observing an expert, or by interacting with the environment. From given examples, machine learning can automatically infer a model that is a formal representation of the structure inherent in the data. The model can be used to predict the environment or to assist humans in understanding the environment. In either case, the learned model requires *reasoning*, that is, making predictions or analyzing how modifying the system's input will change its output. For example, a machine learning system can learn a model based on the medical

records of previous patients. Presented with a new patient record, the system could reason if the patient has a certain disease or not.

Robotics and computer vision are fields that emerged from AI in the early 1970s. They give AI the means to exhibit real-world intelligence by directly manipulating the environment. That is, computer vision gives the machine eyes, or a means of perception, while robotics gives the artificial mind a body, or a means of control and action. It studies ways to turn pixels of images into interpretable concepts, such as objects, scenes, events and beyond, so that computers can understand images in a similar manner as humans do. Thus, computer vision methods acquire, process, analyze, and understand real-world images in order to produce numerical or symbolic information. Computer vision tasks include object tracking, visual recognition and 3D reconstruction. Robotics deals with the construction, operation, and application of robots, as well as systems for their control and sensor processing. Robotics tasks include robot localization, navigation, planning and manipulation.

While classical robotics and computer vision use predefined models (or manual encoding of knowledge) of the robot or its environment for task solving, nowadays, both fields see learning as a central topic. The classical approaches have either proven unsatisfactory in real-world vision tasks, or, although successful for robotic industrial applications, they have fallen behind the more ambitious goal of robotics as a test platform for AI. Machine learning has brought a plethora of advances in extracting statistical models from abstract data in modern computer vision and robotics (similar to speech and bioinformatics). Furthermore, many robotic tasks depend on visual components. For example, robot grasping relies on object recognition, object segmentation and graspable point detection. Thus, the three fields intersect each other and the key contributions of this thesis lie at these intersections.

1.1.2 Visual Recognition

Of all the computer vision tasks, visual recognition is probably the most challenging. Visual recognition consists of analyzing static or dynamic scenes and recognizing its constituent entities. In a dynamic setting the task is one of *recognizing sequences* of interest or events. Following the definition by [Szeliski, 2010], in a static setting, visual recognition can be divided along several axes. It subsumes the *detection* task, which is defined as checking whether a specific element (e.g., a face or an interest point) is in the image and where the match may occur. If the query entity to be recognized is a rigid template then the task is that of instance recognition. The most challenging variant is that of *category recognition* which involves recognizing instances of extremely

varied classes. Visual category recognition may refer to *object recognition*, i.e., naming constituent objects of a certain object category, *scene recognition*, i.e., categorizing an image as belonging to one category of a large range of categories, or *scene understanding*, i.e., naming all constituent objects, their categories and potentially their semantic, spatial and functional relationships. Interlinked in all these tasks is the topic of learning from example images.

This dissertation is concerned with visual recognition. Several visual recognition tasks are essential for robot grasping, others are useful for robot navigation or pre-requisites for truly intelligent artificial agents. For example, detecting good contact points with the robot hand is a critical step for successful object grasping. Object recognition is an important task for robot grasping and navigation. Scene recognition and understanding plays a major role in mobile robot navigation. Finally, recognizing sequences in video data is of central importance in complex dynamic scenes. All these are important visual recognition problems in AI and the main motivation driving this work.

1.1.3 Statistical Relational Learning

Traditional machine learning is concerned with learning from examples represented in an attribute-value (or propositional) format. Propositional representations express knowledge about a single set of properties of the world and do not associate it with objects in the world. For example, in the medical diagnosis case, attributes may be the patient's symptoms, medical record and current medication. In the object grasping domain, attributes can be object shape properties that locally characterize object points.

Although propositional learning has made much progress over the last decades with sophisticated and rigorous statistical techniques yielding accurate models in the presence of noisy data [Bishop, 2006, Szeliski, 2010], in many complex real-world domains a propositional representation is often not appropriate. In the real-world, instances are themselves structured and/or interrelated. For example, in the medical diagnosis problem we may want to also consider the medical records and symptoms of the patient's relatives, but also an explicit family genealogy or relationship. Similarly, for the grasping points we may want to consider properties of neighboring points satisfying certain spatial constraints. In these cases the data exhibits a complex structure and examples are best represented in terms of entities and relationships amongst them.

Furthermore, often real-world problems, such as the ones in computer vision or robotics, cannot rely on complete and precise descriptions of the environment. Thus, the artificial agent should be able to make abstractions and to cope with incomplete or uncertain knowledge. For example, if the goal is to grasp a

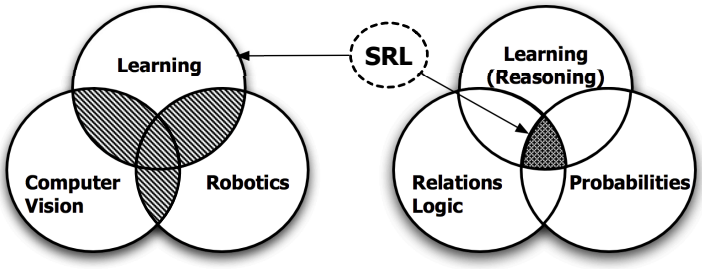


Figure 1.1: This dissertation is situated in three subfields of AI: statistical relational learning (right), computer vision and robotics (left).

cup, then we do not care about its color or the number of handles (as long as there is at least one). If complete descriptions are available, making learning and reasoning effective requires exploiting symmetries and redundancies in the domain, and thus, generalizing over similar situations. These are critical aspects of intelligence not solved yet in computer vision or robotics. This thesis aims to achieve them by means of *relational representations* [De Raedt, 2008], which are most easily described by first-order logic or related formalisms, such as (hyper-) graphs and best supplied by relational languages. When combined with probabilities and statistics, they also provide the possibility to handle uncertainty.

Statistical relational learning (SRL) is an area of machine learning that successfully combines statistical learning and reasoning and relational representations in many complex applications, such as social network modeling, text mining or bioinformatics [Getoor and Taskar, 2007, De Raedt, 2008]. A prominent example are probabilistic logical models that tackle a long standing goal of AI, namely unifying first-order logic –capturing regularities and symmetries– and probabilities –capturing uncertainty. Figure 1.1 shows the general aim of *relational visual recognition*, that is the use of (statistical) relational learning techniques instead of traditional machine learning to solve visual recognition problems for computer vision and robotics.

1.2 Motivation and Research Question

Computer vision and robotics have developed many techniques for visual recognition that use a plethora of local low to medium-level features, including geometric primitives, point clouds, shape and invariant features [Szeliski, 2010].

However, for high-level tasks involving complex objects and scenes such features are not always enough. As examples, consider the tasks of understanding and recognizing individual house facades, distinguishing between restaurant and bar scenes, or finding the best robot grasp based on the object configuration and task-related constraints. In these cases it is the component objects and their complex semantic configuration and interaction that helps recognition. It is more intuitive to understand and describe typical houses as consisting of aligned elements such as a roof, some windows, one or more doors and possibly a chimney. In the bar/restaurant scene example, the differentiating patterns are the consistent qualitative spatial and functional configurations between chairs. One can describe a bar scene as having 'a variable number of chairs of similar size, close to each other and aligned horizontally along a counter'. Finally, high-level reasoning about symbolic object configurations and tasks reduces possible grasps and hence, improves performance. At the same time it allows grasp transfer to novel objects that share similar parts.

In the early days of computer vision, it was felt that hierarchical structure and relations are key components of a scene understanding system [Guzmán, 1968], [Kanade, 1977], [Hanson and Riseman, 1978], [González and Thomason, 1978], [Matsuyama and Hwang, 1985, Fu, 1974]. Popular in early work on syntactic or structural pattern recognition [Haralick, 1983], relational formalisms, such as 'figure description languages' and symbolic graphs, have lost interest in the 1990s [Bunke and Sanfeliu, 1990] due to reasons such as: high computational cost when facing graph complexities, immature low-and mid-level vision features to support such ambitious representations and the limitation of pure relational approaches in handling noisy data. Then, the focus in computer vision was shifted towards low-level representations:

> *"We have showed the use of relational representations, we must yet discover the use of low-level knowledge." Linda Shapiro, 1983*

Furthermore, to perceive, interpret and grasp objects in arbitrary and dynamic environmental scenarios, robot vision capabilities are essential. The majority of grasping methods learns direct mappings from visual perceptions to grasping parameters. However, these methods have a major shortcomming: it is a difficult problem to link gripper parameters to solely local sensor features when dealing with an exploding complexity in the environment and variation in tasks. Only recently, methods that take more global and symbolic knowledge into account have gained more interest. Incorporating domain knowledge (e.g., ontologies) that directly collaborates with the controllers and the (visual) sensors brings increased robustness and can generate more accurate robot grasps.

This thesis wants to contribute towards the idea that visual scenes, grasping scenarios and world knowledge are best described using high-level representational devices that are based on semantically meaningful entities such as graphs, and even more generally using logical and relational languages. We shall argue that the advantages of these rich symbolic representations are: i) they can abstract spatial relations between scene components away from exact locations and thus, generalize over similar situations and view points, ii) they provide means to obtain analytic descriptions of scenes and thus, semantical consistency, iii) they offer contextual knowledge exploitation via symbolic relations, and iv) they transfer knowledge to novel scenarios that share similar semantic entities and thus, generalize over similar (multiple) entities.

Different from early work in computer vision, relational representations have shown robustness to noise when combined with statistical techniques [Antanas et al., 2013a]. Moreover, low-and mid-level vision features are now much more mature. Nevertheless, relational representations have not yet been used to solve visual-based grasping problems or have rarely been used to address visual recognition problems in general (exceptions are grammars for image understanding [Han and Zhu, 2009, Girshick et al., 2011, Zhu et al., 2012], graph mining and rule induction for video data [Sridhar et al., 2010a, Dubba et al., 2010]). Thus, it is time to reconsider old problems with new and successful (statistical) relational learning techniques.

The main research questions of this thesis are *whether visual recognition can benefit from SRL* and *whether we can develop effective and real-world relational visual recognition systems.*

One of the main problems in robotic grasping is generalization across many similar objects and/or tasks. Similarly, one challenge in computer vision is the optimal exploitation of contextual information and generalization across configurations of visual elements. Thus, on one hand, the extraction of similarities between objects and scenes requires relational representations. On the other hand, robotics and computer vision are fields continuously confronted with real-world uncertainties. As a result, we have strong reasons to suspect that SRL techniques can be beneficial for visual recognition tasks in computer vision and robotics.

1.3 Thesis Contribution

We answer these questions via the *key contribution* of this dissertation, which is the use of several (statistical) relational learning techniques for different computer and robot vision problems. This is an important step towards

relational visual recognition and thus, towards closing the loop with the old literature. To achieve this goal, the thesis makes *five main contributions*, three in the field of computer vision and two in the field of robotics. We will now list and describe briefly these contributions.

1. A relational distance-based framework for hierarchical understanding of images. Application: house facades.

Our first contribution is a new relational distance-based framework for hierarchical image understanding. This contribution includes the following:

- a new relational distance function between visual descriptions,
- the use of recent results in relational distance metrics as a relational generalization technique to recognize qualitative high-level structures in images,
- the use of relational generalization throughout all layers of the hierarchy, in a unified way.

2. The employment of a kernel-based relational language for scene classification and scene tagging

Our second contribution is a new relational representation of visual scenes for two important and challenging problems in computer vision: scene classification and scene tagging with object categories. Both problems use a similar relational representation, showing its generality and benefits. Part of this contribution is the employment of the kernel-based language to understand images of houses.

Additional contributions are:

- a high-level relational scene description based on semantic objects and the spatial relationships that hold among them,
- a powerful and expressive representation using (hyper)-graphs,
- a principled way to represent exact metric locations as higher-order relations among objects,
- a deeper insight in scene understanding by employing relations among semantic off-the-shelf object detections.

3. A probabilistic logic pipeline for task-dependent robot grasping

Our third contribution is for robot grasping. It is a new reasoning module based on causal probabilistic logic [Vennekens et al., 2009] and symbolic object parts

for task-dependent robot grasping. Given a set of probabilistic observations about the world, the model can semantically reason about object category, suitable tasks and pre-grasp configurations with respect to the intended task. This contribution comprises:

- the integration of object categorical and task-dependent information for semantic pre-grasp prediction,
- the use of world knowledge about object-task affordances and object/task ontologies to encode general rules that allow generalization over similar object parts and object/task categories,
- a first probabilistic logic module for task-dependent robot grasping.

4. A relational kernel-based approach to numerical feature pooling for robot grasping. Application: graspable point recognition.

Our fourth contribution integrates, using kernels for graphs, numerical appearance features with qualitative spatial relations. Given a 3D point cloud and local shape features of each point, we construct a numerical attributed and symbolic graph by defining spatial relations among points in the cloud. Our goal is to investigate whether the structure of the object can improve graspable point recognition. To achieve it, our approach includes:

- the exploitation of the object graph for extended contextual information,
- the use of spatial proximity to pool numerical shape features.

5. The employment of state-of-the-art SRL systems for video sequence recognition. Application: video streams of Uno *game.*

This last contribution uses sequential statistical relational techniques to capture underlying concepts in video streams. In particular, we focus on monitoring card games and learning to detect fraudulent sequences in Uno video streams. It includes two main steps:

- learning the rules of the Uno game by observing humans playing it from video streams,
- recognizing fraudulent behavior using the learned rules.

Some of the SRL solutions proposed as contributions to the recognition problems considered are framed upon a similar relational kernel-based approach. More

precisely, contributions *2* an *4* rely on the same relational and logical language of a kernel-based framework. They are obtained by changing the relational representation of the problem, while keeping the framework engine. Thus, the SRL approaches proposed are characterized not only by the expressivity of the relational representations, but also by generality with respect to the visual recognition problems addressed. This is an important step towards a general purpose relational visual recognition system.

1.4 Thesis Roadmap

The final part of this introduction gives a brief tour of this thesis. We review robotics, computer vision and statistical and relational learning foundations in **Chapter 2**. The core of the thesis is divided into other three main parts.

Part I is devoted to relational scene understanding and tackles several recognition problems: object recognition, scene recognition and scene understanding. **Chapter 3** provides an insight in the history of syntactic pattern recognition, relations and graphs in visual recognition. Its role is to point out why the popular relational frameworks of the 1970s failed and were abandoned. We discuss what is different now and how SRL can help to solve the old problems. Starting from the trends back then, we overview the recent SRL work for computer vision and point out what would be possible if SRL succeeded. To this aim, **Chapter 4** proposes two new relational approaches to hierarchical image understanding, where the goal is to recognize constituent objects of interest at different levels of semantic granularity. We consider as application the house facades domain. The first approach is a relational distance-based approach which combines robust feature extraction, qualitative spatial relations, relational instance-based learning and compositional hierarchies in one framework. The second approach extends the first one, by replacing the relational distance with a kernel for relational structures. This chapter is based on the following publications:

- Antanas, L., van Otterlo, M., Oramas Mogrovejo, J. A., Tuytelaars, T., and De Raedt, L. *Not far away from home: A relational distance-based approach to understand images of houses.* In Lecture Notes in Computer Science, vol. 6489, pp. 22-29, Inductive Logic Programming, Springer, 2010.
- Antanas, L., Frasconi, P., Tuytelaars, T., and De Raedt, L. *Employing logical languages for image understanding.* In IEEE Workshop on Kernels and Distances for Computer Vision, International Conference on Computer Vision, 2011.

- Antanas, L., van Otterlo, M., Oramas Mogrovejo, J. A., Tuytelaars, T., and De Raedt, L. *A relational distance-based framework for hierarchical image understanding.* In Proceedings of the 1st International Conference on Pattern Recognition - Applications and Methods, 2012, *Best Paper Award.*
- Antanas, L., Frasconi, P., Costa, F., Tuytelaars, T., and De Raedt, L. *A relational kernel-based framework for hierarchical image understanding.* In Lecture Notes in Computer Science, vol. 7626, pp. 171-180, Structural, Syntactic, and Statistical Pattern Recognition, Springer, 2012.
- Antanas, L., van Otterlo, M., Oramas M., J. A., Tuytelaars, T., and De Raedt, L. *There are plenty of places like home: using relational representations in hierarchies for distance-based image understanding.* Neurocomputing Journal, 2013.

In **Chapter 5** we move towards more generic scene understanding where, in a first phase, we contribute a relational kernel-based language for scene recognition. We show that semantic object detections and qualitative spatial constraints between them can improve recognition. In a second phase, we employ a similar relational kernel-based language for scene tagging with object categories. We then iteratively combine object and scene recognition to boost the performance on both tasks. The chapter is based on the following contribution:

- Antanas, L., Hoffmann, M., Frasconi, P., Tuytelaars, T., and De Raedt, L. *A relational kernel-based approach to scene classification.* In Proceedings of Workshop on Applications of Computer Vision, 2013.

Part II discusses SRL techniques for robot grasping. We demonstrate their benefits in **Chapter 6** which proposes a new probabilistic logic pipeline for object grasping, and in **Chapter 7** which presents a new SRL technique to recognize good grasping points in point clouds. The pipeline leverages world knowledge, in the form of object/task ontologies, and low-level data, in the form of point clouds, to improve robot grasping. Starting from a symbolic vision-based scene description, the pipeline first employs a probabilistic logic module to semantically reason about object category, suitable tasks and pre-grasp configurations with respect to the intended task. Once the pre-grasp is determined, the second step in the pipeline maps part-related shape features to good grasping hypotheses. The mapping is done in **Chapter 7** using relational kernels. **Chapter 6** is based on the paper:

- Antanas, L., Moreno, P., Figueiredo, R., Neumann, M., Kersting, K., and De Raedt, L. *High-level reasoning and low-level learning for grasping: a*

probabilistic logic pipeline. Submitted to IEEE Transactions on Robotics, 2013.

Part III focuses on visual sequence recognition with state-of-the-art SRL techniques. The contribution is explained in **Chapter 8** and considers as application UNO card game. The work presented in this chapter has been previously published in:

- Antanas, L., van Otterlo, M., De Raedt, L., and Thon, I. *Learning probabilistic relational models from sequential video data with applications in table-top and card games*. In the Belgian- Dutch Conference on Machine Learning, 2009.
- Antanas, L., Thon, I., van Otterlo, M., Landwehr, N., and De Raedt, L. *Probabilistic logical sequence learning for video*. In Inductive Logic Programming, 2009.
- Antanas, L., Gutmann, B., Thon, I., Kersting, K., and De Raedt, L. *Combining video and sequential statistical relational techniques to monitor card games*. In Proceedings of the ICML Workshop on Machine Learning and Games, 2010.
- Antanas, L., Gutmann, B., Thon, I., Kersting, K., and De Raedt, L. *Combining video and sequential statistical relational techniques to monitor card games*. In Proceedings of the Belgian-Dutch Conference on Machine Learning, 2010.

A **concluding chapter** summarizes the thesis, points out the implications of the results and gives an outlook on future work.

Some of the work performed during my Ph.D research has not been included in the previous chapters. It is either work I am currently investigating and is briefly summarized in **Chapter 9** in the context of related future work, or is listed in the **publication list**.

1.5 Publication List

Journals

- Antanas, L., van Otterlo, M., Oramas M., J. A., Tuytelaars, T., and De Raedt, L. *There are plenty of places like home: Using relational*

representations in hierarchies for distance-based image understanding. Neurocomputing Journal, volume 123, pages 75-85, 2014.

- Janssens, T., Antanas, L., Derde, S., Vanhorebeek, I., Van den Berghe, G., Guiza Grandas, F. *Charisma: An integrated approach to automatic H&E-stained skeletal muscle cell segmentation using supervised learning and novel robust clump splitting.* Medical Image Analysis, volume 17, issue 8, pages 1206-1219, 2013.

Conferences and Workshops

- Antanas, L., Hoffmann, M., Frasconi, P., Tuytelaars, T., and De Raedt, L. *A relational kernel-based approach to scene classification.* In Proceedings of Workshop on Applications of Computer Vision, pages 133-139, 2013.
- Neumann, M., Moreno, P., Antanas, L., Garnett, R., Kersting, K. *Graph kernels for object category prediction in task-dependent robot grasping.* In Online Proceedings of the Eleventh Workshop on Mining and Learning with Graphs, pages 1-6, 2013.
- Billiet, L., Oramas M., J., Hoffmann, M., Meert, W., Antanas, L. *Rule-based hand posture recognition using qualitative finger configurations acquired with the Kinect.* In Proceedings of the 2nd International Conference on Pattern Recognition - Applications and Methods, pages 539-542, 2013.
- Moldovan, B., Antanas, L., Hoffmann, M. *Opening doors: An initial SRL approach.* In Lecture Notes in Computer Science Post Proceedings, Inductive Logic Programming, Springer, pages 178-192, 2013.
- Robben, D., Smeets, D., Ruijters, D., Hoffmann, M., Antanas, L., Maes, F., Suetens, P. *Intra-patient non-rigid registration of 3D vascular cerebral images.* In Lecture Notes in Computer Science, MICCAI Workshop on Clinical Image-based Procedures: From Planning to Intervention, Springer, pages 106-113, 2013.
- Antanas, L., van Otterlo, M., Oramas Mogrovejo, J. A., Tuytelaars, T., and De Raedt, L. *A relational distance-based framework for hierarchical image understanding.* In Proceedings of the 1st International Conference on Pattern Recognition - Applications and Methods, pages 206-218, 2012, *Best Paper Award.*
- Antanas, L., Frasconi, P., Costa, F., Tuytelaars, T., and De Raedt, L. *A relational kernel-based framework for hierarchical image understanding.* In

Lecture Notes in Computer Science, Structural, Syntactic, and Statistical Pattern Recognition, Springer, pages 171-180, 2012.

- Derde, M., Antanas, L., De Raedt, L., Guiza Grandas, F. *An interactive learning approach to histology image segmentation.* In Proceedings of the 24th Benelux Conference on Artificial Intelligence, pages 1-8, 2012.
- Janssens, T., Antanas, L., Derde, S., Vanhorebeek, I., Van den Berghe, G., Guiza Grandas, F. *Charisma: An Integrated Approach to Automatic H&E-stained Skeletal Muscle Cell Segmentation Using Supervised Learning and Novel Robust Clump Splitting Techniques.* In Bioimaging, abstract, 2012.
- Antanas, L., Frasconi, P., Tuytelaars, T., and De Raedt, L. *Employing logical languages for image understanding.* In IEEE Workshop on Kernels and Distances for Computer Vision, International Conference on Computer Vision, pages 1-2, 2011.
- Antanas, L., van Otterlo, M., Oramas Mogrovejo, J. A., Tuytelaars, T., and De Raedt, L. *Not far away from home: A relational distance-based approach to understand images of houses.* In Lecture Notes in Computer Science, Inductive Logic Programming, Springer, pages 22-29, 2010.
- Antanas, L., Gutmann, B., Thon, I., Kersting, K., and De Raedt, L. *Combining video and sequential statistical relational techniques to monitor card games.* In Proceedings of the ICML Workshop on Machine Learning and Games, pages 1-6, 2010.
- Antanas, L., Gutmann, B., Thon, I., Kersting, K., and De Raedt, L. *Combining video and sequential statistical relational techniques to monitor card games.* In Proceedings of the Belgian-Dutch Conference on Machine Learning, pages 1-6, 2010.
- Antanas, L., Thon, I., van Otterlo, M., Landwehr, N., and De Raedt, L. *Probabilistic logical sequence learning for video.* Online Proceedings In Inductive Logic Programming, pages 1-6, 2009.
- Antanas, L., van Otterlo, M., De Raedt, L., and Thon, I. *Learning probabilistic relational models from sequential video data with applications in table-top and card games.* In Proceedings of the Belgian- Dutch Conference on Machine Learning, pages 1-2, 2009.
- Antanas, L., Driessens, K., Croonenborghs, T., Ramon, J. *Using decision trees as the answer network in temporal-difference networks.* In Proceedings of the 18th European Conference on Artificial Intelligence, pages 847-848, 2008.

Chapter 2

Preliminaries

This chapter provides the foundations for the work presented in this thesis. They include the necessary background on robot grasping, computer vision and statistical relational learning. Along the definitions and explanations we will also roughly categorize existing work and thus, provide more context for the contributions. We describe some concepts informally on examples and others more formally following existing literature.

We start by defining fundamental concepts of machine learning that are used throughout this text (Section 2.1.1). Next, we introduce relational data representations in Section 2.1.2. Relational learning and reasoning settings are outlined in Section 2.1.3. Finally, Section 2.2 explains the visual recognition and robot grasping setups in this dissertation.

2.1 Foundations: Machine Learning and Reasoning

This section briefly outlines some fundamental concepts of statistical and relational machine learning and reasoning, and introduces notation and terminology that is used throughout the thesis. More details can be found in [Flach, 2012, Barber, 2011] for statistical machine learning and reasoning and in [De Raedt, 2008] for (statistical) relational learning and reasoning.

2.1.1 Statistical Learning and Reasoning

The general setup in statistical machine learning is based on objects of interest called *instances*. The set of all possible instances is the *instance space* $\mathcal{X}$. Each instance $x \in \mathcal{X}$ is a point in an m-dimensional instance space $\mathcal{X} = d_1 \times \cdots \times d_m$ where d_i is the domain of the i-th attribute describing the input feature x. Instances may have labels and all instance labels together define the output space $\mathcal{Y}$. Both instances and labels can be binary, categorical or continuous. Learning from labeled instances is called *supervised learning*.

We consider supervised learning tasks throughout this thesis. We are given a training set D containing n labelled instances or *training examples* $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. $D \subset \mathcal{X} \times \mathcal{Y}$ is also called *training data* and $e = (x, y)$ a training example. Then the supervised learning task is to find a mapping or a model $\bar{h}$ from the instance space to the output space. It is assumed that examples are independently drawn form a fixed (unknown) distribution P. Such examples are said to be *i.i.d.*. Starting from the type of labels several settings are possible. In this work we focus on *classification*, where labels are binary or categorical, i.e., *classes*. In binary classification it is assumed that $y \in \{-1, +1\}$.

Definition 1. *(Supervised statistical learning).* **Given** *a set of training examples D, a space of possible (probabilistic) classifiers $H = \{h | h : \mathcal{X} \to \mathcal{Y}\}$ and a loss function $L_H : \mathcal{Y} \times \mathcal{Y} \to \Re$,* ***find*** *the classifier $\bar{h} \in H$ with low approximation error $Err(\bar{h})$ on the training data as well as on unseen examples. $Err(h)$ is estimated based on a combination of the training error, e.g., $\frac{1}{n}\sum_{i=1}^{n} L_H(h(x_i), y)$.*

***Example* 1.** *A supervised learning task example is that of patient disease prediction. $\mathcal{X}$ can be the space of all possible patients and $\mathcal{Y}$ is then the space of all possible diagnoses. The i^{th} attribute may be the patient's glucose level at some point in time.*

The classifier above is assumed to be deterministic, that is, it returns a class label $h(x) \in \mathcal{Y}$. A *scoring classifier* h is a mapping from the instance space $\mathcal{X}$ to a k-vector of real numbers $\Re^k$, where $h_i(x)$ is the score assigned to class C_i for instance x. h becomes a *probabilistic classifier* if $h(x)$ is a probability vector over classes, that is $h : \mathcal{X} \to [0, 1]^k$, where $\sum_{i=1}^{k} h_i(x) = 1$. This provides a confidence value for any prediction, allowing further inspection in ambiguous cases.

One of the central paradigms in statistical machine learning is to identify the relevant *random variables* $x \in \mathcal{X}$ from training data, and make a probabilistic model $P(\cdot)$ of their interaction. A discrete probabilistic model (e.g., a

probabilistic classifier) defines a probability distribution $P(\cdot)$ over a set of random variables. The set of assignments a random variable x can take is the domain of x. $P(x)$ denotes the *probability distribution of the random variable* x on all values in its space. *Probabilistic reasoning* is performed by introducing evidence that sets variables in known states, and subsequently computing probabilities of interest of their interaction, conditioned on this evidence. The distribution $P(x)$ can then be used to evaluate a conditional probability distribution $P(x|e) = \frac{P(x,e)}{P(e)}$, called target distribution. In this case x involves random variables and e is the *evidence* or a partial value assignment of the random variables.

***Example* 2.** *In the case of the patient disease prediction example the glucose level is a random variable and its domain is the set of all possible glucose levels. The set of all attributes and the disease target forms the set $\mathcal{X}$ of random variables. At inference time the target distribution is the probability of the disease given the attribute values as evidence.*

The conditional probability distribution $P(y|x)$ expresses a relation between the random variables, that is the probability that the random variable y has a particular value given the knowledge of the evidence x. The random variables can also be related with conjunction instead of condition. $P(x, y)$ is called the joint probability distribution over all possible values of x and y. A *generative* model provides an estimate of the full joint probability distribution $P(x, y)$ on the inputs x and label y. It uses Bayes' rule to calculate $P(y|x)$ and pick the most likely label y. A *discriminative* model provides an estimate of the conditional probability distribution $P(y|x)$ directly or learns a direct map from the inputs x to the class labels y.

In the following we explain the basics of several well-established statistical learning methods used in this thesis: *Support Vector Machines* (SVMs), *k-Nearest Neighbor* (kNN) *Conditional Random Fields* (CRFs) and *n-grams*. However, in our contributions they are upgraded to relational representations. While n-grams are based on generative learning, the other classifiers are discriminative methods.

Support Vector Machines and Kernels. Support Vector Machines (SVMs) are very popular because of their good performance on noisy data and high-dimensional spaces [Boser and et al., 1992, Vapnik, 1995]. We will briefly sketch its principles here, more details can be found in [Shawe-Taylor and Cristianini, 2004, Gartner, 2008]. Assuming a binary classification problem, the goal of a classification method is to find $\bar{h}$ that best separates the two classes.

A linear classifier h is a linear function in the form of a vector of weights w

$$y(x) = sign(\langle w, x\rangle + b), \tag{2.1}$$

where $\langle \cdot, \cdot \rangle$ is the l_2 norm or inner product and b is a constant. Such a linear classifier assumes that the data can be embedded into a space where the separating hypothesis (or hyperplane) is a linear relation $\langle w, x\rangle + b$ in $\mathbb{R}^m$. The examples on one side of the hyperplane are classified as positive, while the others as negative. The Euclidean distance between a point x and the hyperplane is $\frac{|\langle w,x\rangle+b|}{\|w\|}$ where $\| \cdot \|$ denotes the l_2 norm, that is $\| w \| = \sqrt{\langle w, w\rangle}$.

Training a linear classifier is equivalent to finding $\bar{w}$ which constructs a hyperplane (or set of hyperplanes) in the input feature space $\mathcal{X}$ that best interpolates the training set D and can be used for classification:

$$\bar{w} = \arg\min_{w} \frac{1}{n} \sum_{i=1}^{n} L_H(w, x_i, y_i) + \lambda(w), \tag{2.2}$$

where λ is a regularization term that constrains the weights w and L_H is the loss function. At inference time, the prediction for any input x is made using the learned vector $\bar{w}$:

$$y = \arg\max_{y \in Y} (\langle \bar{w}, x\rangle + b), x \in \mathcal{X}. \tag{2.3}$$

If the model is learned using probability estimates, the probabilistic inference step means estimating the target distribution $P(y|x)$.

The ingredients of the SVMs are the maximum-margin principle addressing robustness, slack variables addressing class overlap, and the kernel trick addressing non-linear structure. We first assume a perfect linear classifier and explain the linearly separable examples case. Figure 2.1 (right) depicts such a situation. We then extend it to the non-separable case (slack variables) and non-linear case (kernel trick).

Perfect classification implies that for any x

$$y(x) = \begin{cases} +1 \text{ if } \langle w, x\rangle + b > 0, \\ -1 \text{ if } \langle w, x\rangle + b \leq 0, \end{cases} \tag{2.4}$$

which translates into $\forall x in \mathcal{X},\ y \cdot (\langle w, x\rangle + b) \geq 1$, with the equality holding for the points nearest the hyperplane.

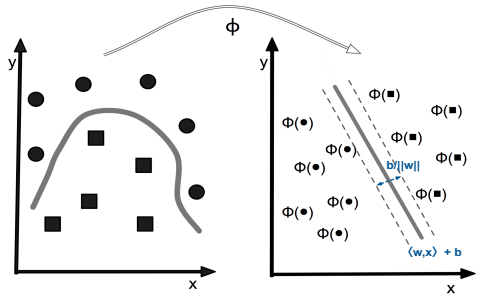


Figure 2.1: The max-margin in SVMs (right). The classes to be separated are -1 (circles) and $+1$ (rectangles). The dotted arrow is the margin. The function ϕ maps the data into a feature space where the nonlinear pattern (left) is now linear (right). The kernel computes inner products in this feature space directly from the inputs.

In a linearly separable setting, there are several perfect classifiers. However, an optimal partition is achieved by the hyperplane that has the largest distance to the nearest training data points of any class or the maximal margin. In general, the larger the margin the lower the generalization error of the classifier. If such a hyperplane exists, it is also known as the maximum margin hyperplane and the linear classifier it defines is known as a max-margin classifier. The margin is illustrated also in Figure 2.1 on the right.

These constraints are cast in the following optimization problem:

$$\begin{aligned} &\min_{w,b} \frac{1}{2}\langle w, w\rangle, \\ &\text{subject to } \forall x \in \mathcal{X} : y_i \cdot (\langle w, x_i\rangle + b) \geq 1. \end{aligned} \tag{2.5}$$

If the data is not separable, SVMs add a tolerance for misclassifications by introducing slack variables ξ in the constraints:

$$\begin{aligned} &\min_{w,b,\xi} \frac{1}{2}\langle w, w\rangle + c\sum_{i=1}^{n} \xi, \\ &\text{subject to } \forall x \in \mathcal{X} : y_i \cdot (\langle w, x_i\rangle + b) \geq 1 - \xi_i,\ \xi_i \geq 0. \end{aligned} \tag{2.6}$$

The positive constant c is the cost parameter of the error term. By increasing c the model misclassifies fewer training examples, but becomes more complex.

The reason why SVMs are so popular is that for most problems, only a limited number of instances lie on the margin. This implies that the expensive computation required by equation (2.2) to find a solution $\bar{w}$ can become sparse, that is features that are not in support vectors get zero weight. This is feasible when the hinge loss function is used for $L_H(w, x, y) = \max\{0, 1 - y \cdot f(x)\} = \max\{0, 1 - y \cdot (\langle w, x\rangle + b)\}$.

Kernel-defined feature mappings. Although the original problem may be stated in a finite dimensional space, it often happens that the classes to discriminate are not linearly separable in that space. Using a feature mapping function $\phi(x)$ instead of x, it is possible to project the original feature space into a higher-dimensional space, presumably making the separation easier in that space (see Figure 2.1). In this context $\phi(x)$ can be either computed explicitly or defined implicitly, via a *kernel function* $k(x, x') = \langle \phi(x), \phi(x')\rangle$. SVMs use a mapping designed via the kernel. The effect is a reasonable computational load based on easily computable dot products in terms of the variables in the original space.

The concept of a kernel formulated as an inner product in the feature space allows any kernel choice. Because the input feature x enters in the form of a scalar product, we can replace that scalar product with the kernel value. In other words, the hyperplane in the higher dimensional space is defined as the set of points whose inner product with a vector in that space is constant. The vector defining the hyperplane is a linear combination with the mapped images of feature vectors in the training set. For more details, see [Bishop, 2007, Schölkopf and Smola, 2002].

Instance-based Learning. SVMs learn an explicit description of the target function (or model) at training time. Differently, instance-based learning is a model-free classifier. It learns an approximation of the real-valued target function $\bar{h}$ by storing the training data. When a new test instance is presented, the set of similar training instances are retrieved and used to classify the new query. As a result, instance-based learning can construct a different approximation to the target function for each test instance. Because the lack of an explicit model, instance-based learning is also called instance-based reasoning.

kNN is an instance-based learning method. It is based on a distance function $d(x, x')$ that measures the difference between two instances x and x'. The most simple distance between two attribute-valued instances is the standard squared Euclidean distance used often in practice. However, other distances are possible as well. Given an instance x, kNN in the basic form assigns to x the most

common class of x's k nearest neighbors, as shown in Equation 2.7.

$$\bar{h}(x) = \arg\max_{y \in \mathcal{Y}} \sum_{i=1}^{k} \delta(y, h(x_i)) \tag{2.7}$$

where x_i are the k nearest neighbors of x, that is with the smallest distances to x; $\delta(y, h(x_i)) = 1$ if $h(x_i) = y$ and $\delta(y, h(x_i)) = 0$ otherwise. In this case the class of x is given by a majority vote of the classes in the training set.

A refined KNN method is to weight the contribution of each of the k nearest examples according to their distance to the query point (Equation 2.8). The class to be picked is the one with the minimum distance. This is called *distance-weighted* KNN.

$$\bar{h}(x) = \arg\max_{y \in \mathcal{Y}} \sum_{i=1}^{k} \frac{\delta(y, h(x_i))}{d(x, x_i)^2} \tag{2.8}$$

KNN can easily be adapted to approximate continuous target value functions and become a scoring or a probabilistic classifier. This can be done by calculating the mean value of the k nearest training examples rather than their most common number. To overcome the high computational cost of KNN to classify new instances, several recent variations aim to obtain a representative training set with a lower size compared to the original one [Garcia et al., 2012]. Such representative training examples are called *prototypes*.

Sequence Classification and Tagging

A particular case of supervised learning is *sequential supervised learning*. It includes tasks such as *sequence tagging* and *sequence classification*. An example is part-of-speech tagging, where one pair (x_i, y_i) might consist of the input sequence $x_i = \langle$do you want water$\rangle$ and the output sequence $y_i = \langle$verb pronoun verb noun$\rangle$. There are several ways to obtain a sequence classifier from a sequence tagger. Chapter 8 employs sequence classification based on sequence tagging to recognize fraudulent playing behavior in game video streams.

Definition 2. *(Sequential supervised learning).* **Given** *a finite set of training examples of the form* $D = \{(x_i, y_i)\}_{i=1}^{n}$, *where each* $x_i = \langle x_{i,j} \rangle_{j=1}^{m}$ *is a*

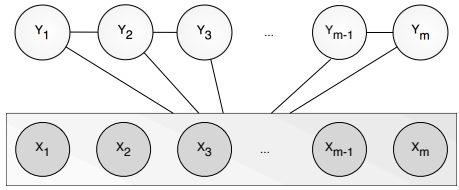


Figure 2.2: Graphical representation of linear-chain CRF.

sequence $\in \otimes\mathcal{X}$ *of elements in the input space* $\mathcal{X}$ *and each* $y_i = \langle y_{i,j}\rangle_{j=1}^{m}$ *is the corresponding sequence* $\in \otimes\mathcal{Y}$ *of elements in the output space* $\mathcal{Y}$, ***find*** *a function* $\bar{h} : \otimes\mathcal{X} \to \otimes\mathcal{Y}$ *with low approximation error* $Err(\bar{h})$ *on the training data as well as on unseen examples.*

Conditional Random Fields. CRFs are popular representations for sequential supervised learning [Lafferty et al., 2001]. They are undirected graphical models that encode a conditional probability distribution using a given set of features. As a special case, consider a linear-chain CRF, where $x = \langle x_1 \dots x_m\rangle = \langle x_i\rangle_{i=1}^{m}$ and $y = \langle y_1 \dots y_m\rangle = \langle y_i\rangle_{i=1}^{m}$, so that y is a labeling of an observed sequence x. Then, CRFs define the conditional probability of a state sequence given the observed sequence as

$$P(y|x) = Z(x)^{-1} \exp \sum_{t=1}^{m} \Psi_t(y_t, x) + \Psi_{t-1,t}(y_{t-1}, y_t, x)$$

where $\Psi_t(y_t, x)$ and $\Psi_{t-1,t}(y_{t-1}, y_t, x)$ are potential functions and $Z(x)$ is a normalization factor over all state sequences $y \in \mathcal{Y}$. A *potential function* is a real-valued function that captures the degree to which the assignment y_t to the output variable fits the transition from y_{t-1} and x. Due to the global normalization by $Z(x)$, each potential has an influence on the overall probability.

To apply CRFs to sequential supervised learning problems, one must choose a representation for the potentials. Typically, it is assumed that the potentials factorize according to a set of features $\{f_k\}$, which are given and fixed, so that $\Psi(y_t, x) = \sum \alpha_k g_k(y_t, x)$ and $\Psi(y_{t-1}, y_t, x) = \sum \beta_k f_k(y_{t-1}, y_t, X)$ respectively. The model parameters are now a set of real-valued weights α_k, β_k, one weight for each feature. In linear-chain CRFs, a first-order Markov assumption is made on the random variables. A graphical model for this is shown in Figure 2.2. In this case, there are features for each label transition. Feature functions can be arbitrary such as a binary test that has value 1 if and only if y_{t-1} has the correct label. As one can see, every output node depends on the complete input.

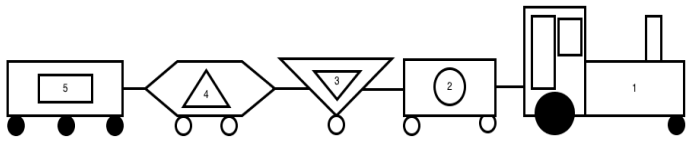


Figure 2.3: How would you characterize this train? A possible way is by "every second car that is not an engine and has the shame shape as its cargo". This rule can distinguish this train from others that do not have similar structure and properties.

N-grams. Propositional n-grams [Manning and Schütze, 1999] estimate the probability of a sequence $x = \langle x_i \rangle_{i=1}^{m}$ as smoothed Markov chains, a finite mixture of Markov distributions of different orders. A Markov chain of order n estimates the probability of x as follows

$$P_n(x) = \prod_{i=1}^{m} P_n\big(x_i | x_{i-n} \dots x_{i-1}\big) = \prod_{i=1}^{m} \frac{C(x_{i-n} \dots x_i)}{C(x_{i-n} \dots x_{i-1})} \tag{2.9}$$

where the conditional probabilities are estimated from a set D of training sequences using 'gram' counts: $C(x_1 \dots x_k)$ is the number of times $\langle x_1 \dots x_k \rangle$ appeared as a subsequence in any $x \in D$. To avoid overfitting of the model for a large gram order n, models of different orders can be combined. Consequently, the conditional probabilities are then defined as

$$P_n(x_i | x_{i-n} \dots x_{i-1}) = \sum_{k=1}^{n} \alpha_k P_k\big(x_i | x_{i-k} \dots x_{i-1}\big) \tag{2.10}$$

where $\alpha_1, \dots, \alpha_n$ are positive weights with $\sum_{k=1}^{n} \alpha_k = 1$ and P_k is the conditional distribution defined by a k-order gram. N-grams estimate a model $P(x)$ for each class label y. At prediction time the model is used to calculate the posterior probability $P(y|x) \propto P(y) \cdot P(x|y)$.

2.1.2 Relational Data Representations and Learning

The machine learning concepts described above are based on traditional propositional representations that work with fixed feature vectors or attribute-

value representations. However, consider the train in Figure 2.3[1]. It is a structured object consisting of a sequence of cars, each of different shapes and sizes and containing differently shaped cargos. Each car, depending on the shape and size, can have different number of wheels. Such an instance is best described in terms of its constituent objects, their properties and their dependencies. *Relational* representations represent a variable number of entities and relationships amongst them. In fact, the real world has the form of complex and structured data collections and is best represented in terms of relational representations. Complex data representations include *relational databases* and *graphs*. Graph structures are found whenever data consists of individual entities connected by links. A real-world graph example of an indoor scene from Chapter 5 is given in Figure 2.4. Using relational representations more general structures such as hypergraphs can also easily be represented.

The main idea behind relational representations is to organize data in relations, that is, *sets of tuples*. There are two major formalisms: the entity-relationship (E/R) model from databases and first-order logic. When derived from first-order logic, relational representations are called *logical* representations. There is a straightforward mapping from the E/R models used in relational databases to first-order logic [Das, 1992], meaning that any relational database can be converted to a logical representation. They are both used throughout this thesis, therefore we review and exemplify their key elements.

E/R model

The classic entity/relationship (E/R) data model [Garcia-Molina et al., 2008] used in database theory is represented graphically as an *entity-relationship (E/R) diagram* using three principal element types: entity sets, attributes and relationships.

An *entity* is an abstract object of some sort and a collection of similar entities forms an *entity set*. An entity resembles an "object", which is a static concept involving structure of the data.

***Example* 3.** *(Relational train example). Let us consider the train example in Figure 2.3. We can build a train database were the entities are cars, parts of cars (e.g., wheels, windows) and cargos.*

Relationships are connections among two or more entity sets. For instance in the train example a contain relationship between the cars and the cargos is possible. Entity sets and relationships can have associated *attributes*, which are

[1] This example is based on [Michalski et al., 1986].

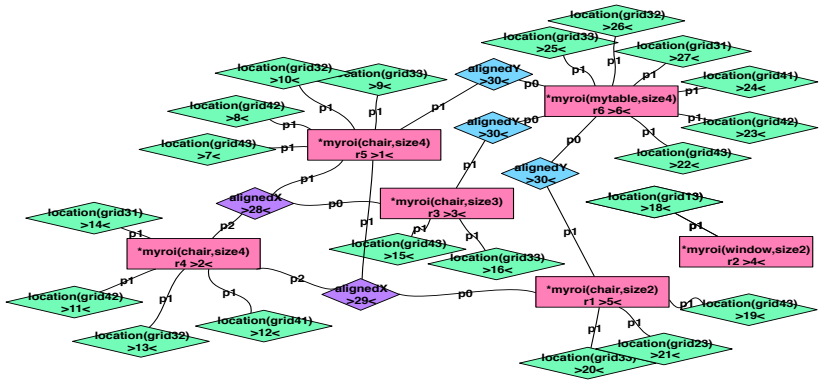


Figure 2.4: (Partial) graph of a real-world dining room visual scene. Rectangles are entities, diamonds are relationships among entities and they have properties.

properties of the entities in that set or of the relationships. Attributes are of primitive types, such as categorical (or discrete), integer or real. For instance, the entity *car* has an attribute indicating if the car is an engine or not and the shape of the car.

An E/R diagram is a graph representing entity sets, attributes and relationships. Elements of each of these are represented by nodes of the graph, as follows: entities are represented by rectangles, attributes by ovals and relationships by diamonds. Edges connect an entity set to its attributes and a relationship to the entity sets it links. Figure 2.5 illustrates an E/R diagram that represents our train database example. E/R diagrams are a notation for describing schemas of databases or a visual interpretation of the database. When the diagram is grounded with particular data we obtain an "instance" of the database. The E/R diagram lists as entities *car* with properties *id* (the identifier of the car), *ctype* (the type of the car, i.e., engine or not) and *shape* (the shape of the car), *cargo* with properties *id* (identifier of the cargo) and *shape* (the shape of the cargo), and *carPart* with properties *id* (the identifier of the part), *ptype* (the type of the part, i.e., wheel, window, engine chimney) and $fill$ (if the part is black or white). Relationships involved in the E/R diagram are *contains* between *cargo* and *car* entities, *attached* between *part* and *car* entities and *linked* (behind) between two *cargo* entities.

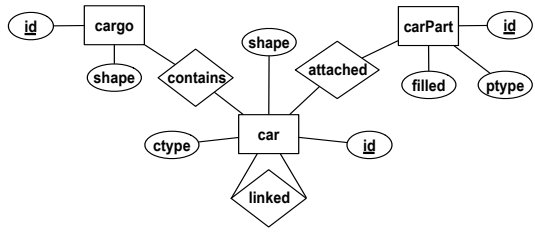


Figure 2.5: E/R diagram for the train domain example.

Logic

An atom is an expression of the form $\mathtt{p}(t_1, \ldots, t_k)$ where $\mathtt{p}/k$ is a predicate symbol of arity k and t_i are *terms*. Terms are either *constants*, *variables* or other structured terms of the form $f(t_1, \ldots, t_k)$ (or *functors*)[2]. Constants are denoted in lower case and variables in upper case. Ground expressions do not contain variables. Terms are arguments of atoms. Ground terms represent entity identifiers or properties. Ground atoms represent particular relations, i.e., entities and/or relationships. Also, ground atoms possess truth-values, that is they are either true or false. When ground atoms are true they are also called *facts*. Negated atoms have the form $not(\mathtt{p}(t_1, \ldots, t_k))$. Using these notions we can now define a *clause* q as an expression of the from $h_1; \ldots; h_n \leftarrow b_1, \ldots, b_m$, where h_i and b_j are logical atoms. The arrow means implication (if). The right side of the arrow, denoted $body(q)$, is the *body* of the clause and it is a conjunction of atoms. The left side, denoted $head(q)$, is the *head* of the rule and it is a disjunction of atoms. Furthermore, all variables are universally quantified.

***Example* 4.** *The rule*

$$\mathtt{engine(X); cargoCar(X)} \leftarrow \mathtt{car(X)}$$

specifies that for all X, when X is a car, X is an engine or a cargo car.

From this general definition, we can obtain particular cases of definite clauses or facts if $n = 1$, and $n = 1$ and $m = 0$, respectively. In the logic formalism the same database specified with the E/R diagram consists of a set of clauses which specify that all the clauses are true. In our train example we use the predicate symbols `car`/3, `cargo`/2, `part`/3, `contains`/2, `attached`/2, `linked`/2, while *rectangle*, *circle*, *triangle* are constants. A predicate which does not contain

[2]In practice, we use a functor-free language.

any variables is, for example, $car(2, cargo\_car, rectangle)$. A *substitution* $\theta = \{V_1/t1, \ldots, V_k/t_k\}$ is an assignment of terms $t_1, \ldots, t_k$ to the variables $V_1, \ldots, V_k$. We obtain an instantiated formula, denoted $q\theta$, if all variables in q are replaced by the terms in θ.

An important feature of learning in logic-based representations is the possibility to employ *background knowledge*. Background knowledge is any form of prior knowledge relevant for the problem at hand which is assumed to be correct a priori. In our train example it can be the fact that any two consecutive cars in a train are connected. Background knowledge is represented as a set of definite clauses, and it is called also *background theory*.

The semantics of a database is strongly linked with the concept of *interpretation*. An interpretation of a domain of a set of clauses is an assignment that maps constants onto the entities in the domain, predicates p/k onto k-ary relations defined in the domain (or a set of k-tuples objects) and function symbols f/k onto k-ary functions defined on the domain. It defines which of the possible atomic statements are true in a given interpretation. Example 5 illustrates an interpretation for the train example. More formally,

Definition 3. *A (Herbrand)* **interpretation** *of a set of clauses is a set of ground atoms over the set of predicate symbols, constant symbols, and functor symbols that occur in the set of clauses considered.*

***Example* 5.** *(Relational train example). An interpretation can be the following:* $i = \{\texttt{car}(1, engine, complex), \texttt{car}(2, ccar, rectangle), \texttt{car}(3, ccar, triangle), \texttt{car}(4, ccar, hexagon), \texttt{car}(5, ccar, rectangle), \texttt{cargo}(6, circle), \texttt{cargo}(7, triangle), \texttt{cargo}(8, triangle), \texttt{cargo}(9, rectangle), \texttt{part}(10, wheel, black), \texttt{part}(11, wheel, black), \ldots, \texttt{part}(20, wheel, black), \texttt{part}(21, window, white), \texttt{part}(22, window, white), \texttt{contains}(2, 6), \texttt{contains}(3, 7), \texttt{contains}(4, 8), \texttt{contains}(5, 9), \texttt{attached}(10, 1), \texttt{attached}(11, 1), \ldots, \texttt{attached}(20, 5), \texttt{attached}(21, 1), \texttt{attached}(22, 1), \texttt{linked}(1, 2), \ldots, \texttt{linked}(4, 5).\}$

A (Herbrand) interpretation I is a model of a clause q if and only if for all substitutions θ such that $body(q)\theta \subset I$ is true, it is also true that $head(q)\theta \subset I$. I is a model of a set of clauses if it is a model of all clauses in the set.

Relational Learning and Generality

As for propositional data, different supervised machine learning methods can be considered for relational data. Examples of relational tasks include classifying entire graphs (e.g., visual scenes), or individual nodes within a graph (e.g., objects in a scene). Chapters 4 and 5 contain concrete examples of such

prediction problems. From a logical perspective prediction tasks are similar in the sense that they all involve graph nodes that correspond to tuples of the form $p(t_1, \ldots, t_k)$. When learning from relational data one possible setting is that of *learning from interpretations*. In this case, examples are complete interpretations and provide, in this setting, more information than when learning from single clauses (or facts) as examples[3].

Definition 4. *(Learning from interpretations).* **Given** *a background theory B, a set of training examples D in the form of interpretations, a set of clauses $\mathcal{H}$ that specifies the clauses that are allowed in hypotheses,* **find** *a hypothesis $h \in \mathcal{H}$ such that h covers all positive training examples in D and maximizes a score function $s(D, h, B)$ specifying the quality of h w.r.t. the data D and the background theory. The hypothesis h covers a training example e if e is a model of $B \cup h$.*

Relational learning is based on the idea of *generality* , defined as follows

Definition 5. *(Generality). Let h_1, $h_2 \in \mathcal{H}$. Hypothesis h_1 is more general than hypothesis h_2 ($h_1 \preceq h_2$) if and only if all examples covered by h_2 are also covered by h_1.*

There are several frameworks used for generalization. One of them is $\Theta -$ *subsumption* with subsumption at the level of logical atoms. Clause q_1 $\Theta -$ *subsumes* clause q_2 if and only if $\exists$ a substitution θ such that $q_1\theta \subseteq q_2$. In this case, q_1 is more general than q_2. Example 6 shows two cases of Θ-subsumption. In the second example a longer clause can subsume a shorter clause, a difficulty which one may want to avoid. This occurs when there exist substitutions θ for which several literals in the longer clause collapse into one.

***Example* 6.** *The clause $c_1 = \mathtt{q(a)} \leftarrow \mathtt{p(a), r(a)}$ is Θ-subsumed by the clause $c_2 = \mathtt{q(X)} \leftarrow \mathtt{p(X), r(X)}$ with substitution $\theta = \{\mathtt{X/a}\}$.*

The clause c_1 is also Θ-subsumed by the clause $c_3 = \mathtt{q(X)} \leftarrow \mathtt{p(X), p(Y), r(X)}$ with substitution $\theta = \{\mathtt{X/a, Y/a}\}$. In this case, the literals $\mathtt{p(X)}$ and $\mathtt{p(Y)}$ collapse, after the substitution, into $\mathtt{p(a)}$.

A variant of Θ-subsumption is the *object identity* ($\mathcal{OI}$) subsumption. It prevents collapsing literals because it does not allow to substitute two different terms with the same variable or the same term with two different variables. In other words, it requires that all terms in the same clause are different. More formally, clause q_1 $\mathcal{OI} -$ *subsumes* clause q_2 if and only if there is a substitution θ such that $\mathit{completed}(q_1)\theta \subseteq \mathit{completed}(q_2)$, where $\mathit{completed}(q) = q \cup \{t_i \neq t_j | t_i$ and t_j are two distinct terms in $q\}$.

[3]This second setting is called *learning from entailment.*

***Example* 7.** *Given the clause* $c_3 = \mathtt{q(X)} \leftarrow \mathtt{p(X), p(Y), r(X)}$*,* $completed(c_3)$ *is* $\mathtt{q(X)} \leftarrow \mathtt{p(X), p(Y), r(X), X} \neq \mathtt{Y}$.

The clause $c_1 = \mathtt{q(a)} \leftarrow \mathtt{p(a), r(a)}$ *is both* Θ*-subsumed and* $\mathcal{OI}$*-subsumed by the clause* $c_2 = \mathtt{q(X)} \leftarrow \mathtt{p(X), r(X)}$ *with substitution* $\theta = \{\mathtt{X/a}\}$*. However,* c_1 *is not* $\mathcal{OI}$*-subsumed by* c_3 *due to the constraint* $\mathtt{X} \neq \mathtt{Y}$ *in* $completed(c_3)$*.*

Based on the idea of generality and generalization operations we can define two concepts that are useful for learning in general, and in our case to explain the notion of relational distance. The *minimally general generalization* of two hypotheses h_1 and h_2 is defined as $\text{mgg}(h_1, h_2) = min\{h \in \mathcal{H} | h \preceq h_1 \text{ and } h \preceq h_2\}$, where $\preceq$ indicates generality up to equivalence. The mgg is unique if calculated in the Θ-subsumption framework. In this case the mgg is called *least general generalization* and denoted lgg. The mgg represents computing the lgg under $\mathcal{OI}$-subsumption [Khoshafian and Copeland, 1986, Semeraro et al., 1996] and it may not be unique. Illustrations of mggs under $\mathcal{OI}$-subsumption and Θ-subsumption are shown in Example 8.

***Example* 8.** *Let* $h_1 : \mathtt{q(e)} \leftarrow \mathtt{p(a), r(a)}$ *and* $h_2 : \mathtt{q(f)} \leftarrow \mathtt{p(b), r(c)}$.
Under Θ*-subsumption there is one mgg:*
$\text{mgg}_\Theta = \{\mathtt{q(Z)} \leftarrow \mathtt{p(X), r(Y)}\}$ *with* $\theta_1 = \{\mathtt{X/a, Y/a, Z/e}\}$, $\theta_2 = \{\mathtt{X/b, Y/c, Z/f}\}$.
Under $\mathcal{OI}$*-subsumption there are two possible* mgg*s:*
$\text{mgg}^0_{\mathcal{OI}} = \{\mathtt{q(Y)} \leftarrow \mathtt{p(X)}\}$ *with* $\theta^0_1 = \{\mathtt{X/a, Y/e}\}$, $\theta^0_2 = \{\mathtt{X/b, Y/f}\}$
$\text{mgg}^1_{\mathcal{OI}} = \{\mathtt{q(Y)} \leftarrow \mathtt{r(X)}\}$ *with* $\theta^1_1 = \{\mathtt{X/a, Y/e}\}$, $\theta^1_2 = \{\mathtt{X/c, Y/f}\}$.

2.1.3 Statistical Relational Learning and Reasoning

In real-world domains, systems need to handle both inherent uncertainty and relational structure. Consider the train example, which assumes perfect shapes. Real-world shapes, however, are much noisier. In turn, by restricting their attention to simple attribute-value representations, propositional machine learning approaches ignore much of the relational complexity of the real world. Statistical machine learning easily handles uncertainty, while inductive logic programming and related techniques address learning in relational domains. Statistical relational learning combines these two directions of research into techniques that perform statistical learning as well as reasoning in relational domains.

The integration of statistical and relational learning/reasoning is approached from different directions in this thesis. One perspective is to start from statistical techniques and extend them to relational data representations. This perspective is implemented by *structured SVMs* and *kernels for graphs*, kNN with *relational distances* and *graphical models lifted to logical representations*. The other

perspective is to start from relational learning techniques and extend them with probabilities. It includes probabilistic extensions of logic languages. One probabilistic logic is Causal Probabilistic (CP) Logic (or *CP-logic*) [Vennekens et al., 2009]. These techniques are used throughout this thesis and we briefly introduce them here, with more explanations in their respective chapters.

Relational Distances. There are a wide range of possible distance measures and metrics for structured data, either based on decomposition or based on generalization. In this thesis we consider a relational distance based on generalization which tries to find a common part among two interpretations. The distance, introduced in Chapter 4, is based on a generalized metric result [De Raedt and Ramon, 2009] which is very useful to construct relational distance functions and which we briefly introduce. The main idea is that one can build a metric based on the notion of mgg introduced in Section 2.1.2 for any partially ordered hypothesis space $\mathcal{H}$ under some mild assumptions. More precisely

Definition 6. *(Generalization metric). Assuming a partially generality order on a hypothesis space $\mathcal{H}$, an anti-monotonic and strict order preserving* size function $|.|$ *and a defined minimally general generalization* mgg *of two hypotheses h and h' (yielding at least one element), then the function*

$$d(h, h') = |h| + |h'| - 2 \cdot |\operatorname{mgg}(h, h')|, \forall h, h' \in \mathcal{H} \tag{2.11}$$

is a **distance metric**.

The result allows to derive distance metrics for different types of instances, including hypergraphs, and therefore can be used to compute distances between interpretations. For example, one can choose a *graph isomorphism* as a partially ordered generality relation which induces a generality order on graphs, where the size function can be the number of vertices [Bunke and Shearer, 1998]. In this case the mgg corresponds to the *maximal common subgraph*. Computing the distance between graphs g_1 and g_2 is equivalent to calculating the distance between their corresponding interpretations i_1 and i_2 using $\operatorname{mgg}(i_1, i_2)$.

The lgg could also be used to find a *common part* between interpretations (resp. graphs). However, it allows for different variables in the lgg to unify. This collapse of literals into one would violate the strictly ordering preserving condition for the size function. The mgg on the other hand is not unique (we can find multiple common parts) and is the result of exact *structure matching*, i.e., each constant in an interpretation (resp. each node in a graph) must be matched against a different constant (resp. node) in the other. Exact structure

matching makes also more sense in computer vision problems, since we want to find specific structures and do not want for example to collapse two objects in a scene into one.

Kernels for Structured Data. It addresses the setting in which the input instances to the kernel function are rich compositions, that is, they can be interpretations or graphs. Kernels for structured data can be constructed starting from the decomposition kernel paradigm [Haussler, 1999] based on *parts* and *decomposition relations*. If $x \in \mathcal{X}$ is a structured instance, such that it can be decomposed into parts $(x_1, \ldots, x_P)$, and there is a decomposition relation $R(x, x_1, \ldots, x_P)$ such that x_i are parts that belong to x, the decomposition kernel $\kappa(x, x')$ is:

$$\kappa(x, x') = \sum_{\substack{x_i \in R^{-1}(x) \\ x'_j \in R^{-1}(x')}} \kappa_p(x_i, x'_j) \tag{2.12}$$

where $R^{-1}(x)$ returns the parts of x, that is $R^{-1}(x) = \{(x_1, \ldots, x_P) : R(x, x_1, \ldots, x_P)\}$ and κ_p denotes a kernel on parts. For kernels on vectors, parts are attributes of the vector, while in the case of graphs parts may be paths in the graph, subgraphs [Gärtner et al., 2004] or pairs of subgraphs [Costa and De Grave, 2010]. Kernels for structured data are conveniently defined to calculate explicit feature mappings $\phi(x)$ from interpretations to attribute-value vectors. Computing a direct mapping offers advantages when dealing with large scale learning problems (with many interpretations) and structured output tasks (exponentially many possible predictions), as one can directly control the complexity of the extracted features.

CP-logic. Similar to other probabilistic logics, CP-logic aims to evaluate a target distribution $P(x|e)$, where x is a query involving random variables and e is the evidence. In a relational setting, random variables often correspond to ground atoms, such that $P(x|e)$ defines a distribution over truth value assignments. A CP-theory is a set of CP-events or rules of the following form:

$$(\alpha_1 :: h_1; \alpha_2 :: h_2; \ldots; \alpha_n :: h_n) \leftarrow b_1, \ldots, b_m$$

where α_i are causal probabilities, $\alpha_i \in [0, 1], \sum_{i=1}^{n} \alpha_i \leq 1$. In this case the head of the rule is a set of pairs $(\alpha_i : h_i)$. We also refer to the h_i as consequences, and to the b_i as conditions. Each rule may have several possible consequences

in its head, and each consequence h_i has a causal probability α_i assigned to it. If the body of the rule is true, then the rule makes at most one of these consequences true. The probability that the rule causes h_i to be true is α_i. Thus, the probability should be interpreted as: if the body is true, then it causes the consequence to become true with a causal probability. In any CP-rule, the sum of the possible outcomes can be at most 1. If the sum is less than 1, there is a non-zero probability that nothing happens.

***Example* 9.** *(Meal cooking). Let us consider the scenario in which a robot may prepare a dish and it chooses to use pasta or rice. This can be written in CP-logic as*

$$0.6 :: makes(dish).$$
$$0.5 :: used(rice); 0.5 :: used(pasta) \leftarrow makes(dish).$$

A random variable in this case is the atom $used(rice)$*. The rules express that the robot may cook a main dish with probability* 0.6 *and if it does it can be pasta or rice, each with* 50% *chance. The fact that the robot makes a dish causes the existence of a pasta meal or a rice meal, but not both. We can extend the model with another type of meal. For example, the robot may make also a soup and we assume that it could make both meals. If it makes a soup, then it may use either rice or vermicelli, but not both.*

$$0.3 :: makes(soup).$$
$$0.3 :: used(rice); 0.7 :: used(vermicelli) \leftarrow makes(soup).$$

This results in several rules in the CP-theory that may lead to a same consequence: if the robot makes both a main dish and a soup it is possible that it uses rice in both.

The aim of CP-logic is to offer a formal language to model causal knowledge. It incorporates dynamic concepts, such as events and processes. The fundamental kind of information is why events occur and what are the effects of events. It is part of the semantics of CP-logic that each rule independently of all other rules makes one of its head atoms true when triggered. CP-logic is therefore particularly suitable for describing models that contain a number of independent stochastic events or causal processes.

The interpretation of the probability in the head is different from the conditional probability that the head is true given the body. It reflects the probability that the body causes the head to become true. Among the two, the former is local knowledge: an expert can estimate the probability that $makes(soup)$ causes

$used(rice)$ without considering any other possible causes for $used(rice)$. To infer $P(used(rice)|makes(soup))$, we need global knowledge: we need to know all possible causes for $used(rice)$, the probability of them occurring, and how they interact with $makes(soup)$. More explanations on the meaning of CP-logic with examples for robot grasping are given in Chapter 6.

***Example* 10.** *(Meal cooking).*
It may be tempting to interpret $P(used(rice)|makes(soup)) = 0.3$*, however the conditional probability that rice is used, given that a soup is made, is higher because there is a second possible cause for using rice, that of cooking a main dish. Thus,* $P(used(rice)|makes(soup)) = 0.3 + 0.7 \cdot 0.6 \cdot 0.5 = 0.51$*. The rice is used for the soup with probability* 0.3*, but there is also a probability* $1 - 0.3$ *that the soup is not made. In this case, it is possible that a main dish is cooked with probability* 0.6 *and rice is used for it with* 0.5 *probability.*

There are several queries that we can ask the CP-theory. For example, the probability that pasta is used: $P(used(pasta)) = 0.6 \cdot 0.5 = 0.3$*. Pasta can be used with probability* 0.5 *only when a main dish is cooked, thus, with probability* 0.6*. Similarly, we can calculate the probability of vermicelli being used:* $P(used(vermicelli)) = 0.7 \cdot 0.3 = 0.21$*. The probability that rice is used is triggered by two rules and is calculated using the noisy-or formula:* $P(used(rice)) = 1 - (1 - 0.5 \cdot P(makes(dish)) \cdot (1 - 0.3 \cdot P(makes(soup)) = 1 - (1 - 0.5 \cdot 0.6) \cdot (1 - 0.3 \cdot 0.3) = 0.363$*.*

2.2 Background: Visual Recognition and Robot Grasping

This section briefly explains the context of this thesis and gives some notes on the terminology from the application point of view. For more detailed explanations please see [Sahbani et al., 2012] for robot grasping and [Tuytelaars and Mikolajczyk, 2007, Szeliski, 2010] for computer vision notions. The goal of this section is not to give a exhaustive explanation of all existing interest point detectors, descriptors or recognition methods, but to help later explanations on visual primitives or baselines used.

2.2.1 Local Features, Points and Regions of Interest

Local feature points, also denoted local features or image patches, are the basic concept of many advances in computer vision applications. A local feature is an image pattern which differs from its immediate neighborhood, which means

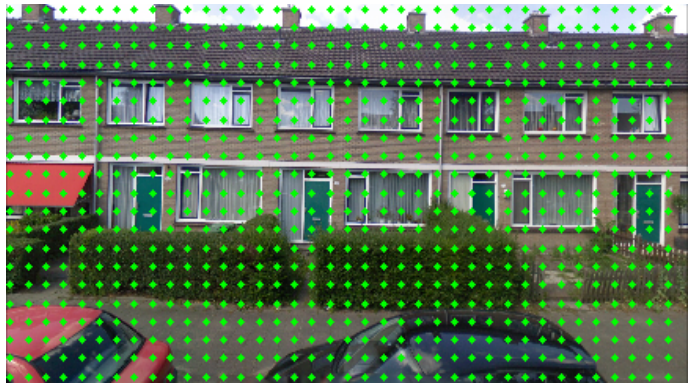
that it is often caused by significant local change(s) of image properties. Local features can be very broadly divided into two different types of representations: interest points and dense sampling.

Dense sampling is performed on a regular grid and results in a good coverage of the image with a constant amount of features per image area. Each sample is associated to a local feature. Different from densely sampled points, *interest points* correspond to a well-defined and reproducible concept across different images. Usually, all interest points are characterized by some implicit spatial extent beyond the point location itself. When the interest point needs to be described by a local neighborhood of pixels in order to be used for further processing, it becomes a *region of interest*. Thus, interest point detectors focus on 'interesting' regions, which are typically regions with high information content, that can be localized precisely. They are also often called feature detectors. Figure 2.6 exemplifies the types of local features.

The number of interest points extracted from an image varies depending on the image content. Interest points have proved useful when the goal is to find correspondences between object categories or between scenes categories. Furthermore, there are interest points that are *invariant* to scale, rotation and affine transformations. Examples of interest points detectors include geometric primitives detectors (e.g., edges, curvatures), gradient-based detectors (e.g., Harris corner detector, SIFT) and intensity-based detectors.

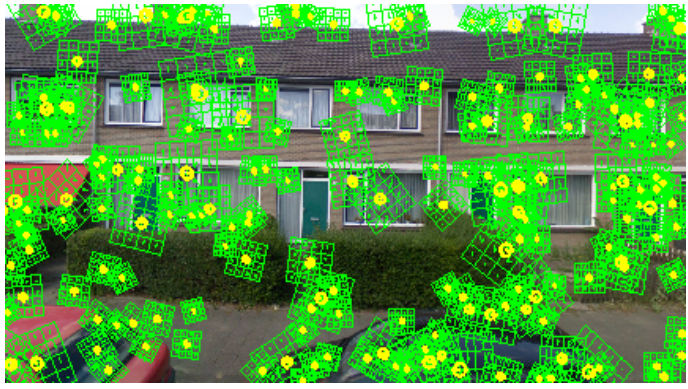
k Adjacent Segments

An interest point detector that we employ in this work is *k adjacent segments* (or kAS) [Ferrari et al., 2008]. kAS are contour-based local features that group adjacent (or connected), approximately straight contour segments. The segments in a kAS form a path of length k through a network of contour segments that covers the image. Two segments are connected if they are adjacent on the same contour or if one segment is at the end of another contour and directed towards the other segment. The larger the number of k, the more complex the local feature. We obtain for $k = 1$ just edges, for $k = 2$ 'L'-like shapes (or corners) and, for $k = 3$, shapes such as 'Z' and 'F'. Except being interest points, one can extract feature descriptors using the contour network. Figure 2.6(b) shows 2AS interest point detections on a street view image.



(a) Dense sampling on a regular grid using a sampling step of 50 pixels.

(b) Interest point detection using 2AS. The circles mark corners and the colored dots contours found in the image.



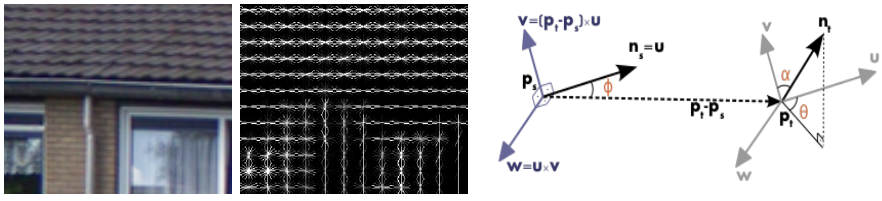
(c) Interest points characterized by SIFT descriptors calculated in their neighborhoods (regions of interest) using 4x4 grids. The interest points are obtained using differences of gradients.

Figure 2.6: Examples of dense sampling, interest points and regions of interest on a house facade image.

2.2.2 Feature Descriptors

After detecting interest points, in order to be compared for recognition, they are characterized by *feature descriptors*. Feature descriptors are obtained by extracting a numerical vector that characterizes locally the image patch around the interest point. There exist a large number of descriptors which emphasize different image properties, depending on the task at hand [Mikolajczyk and Schmid, 2005]. Many of them are based on histograms of certain (mostly geometric) properties in the smoothed patch. There is a plethora of local feature descriptors. Some examples of 2D local feature descriptors include shape context [Belongie et al., 2002], SIFT [Lowe, 2004] (see Figure 2.6(c)), SURF [Bay et al., 2006], HOG [Dalal and Triggs, 2005]. Local descriptors developed for 3D images include 3D SURF, 3D shape context and point feature histogram [Rusu et al., 2009].

We further note down the difference between local and global descriptors. While local descriptors are calculated locally on an image region and summarize local (geometric) changes, global descriptors consist of feature vectors describing the image content. Examples of global feature descriptors are GIST [Oliva and Torralba, 2001] and color histograms, among others. Because of the difference in semantic content between object and scene recognition, the results obtained with local descriptors is often quite different from those obtained with global descriptors for the same task. Exceptions are, for example, bags of words and spatial pyramids which are very popular in both object and scene recognition [Fei-Fei and Perona, 2005, Grauman and Darrell, 2005, Lazebnik et al., 2006, Pandey and Lazebnik, 2011]. We present in more detail some of the descriptors employed in our work.



(a) (b)

Figure 2.7: Illustration of the HOG descriptor (a) and PFH computation [Rusu and Cousins, 2011] (b).

Point Feature Histograms

The *point feature histogram* (or PFH) encodes the statistics of the shape of a depth image (or point cloud) by accumulating the geometric relations between all the point pairs. Given a pair of points in the neighborhood and their normals (see Figure 2.7(b)), the PFH accumulates the four dimensional histogram of: the cosine of the angle α, the cosine of the angle ϕ, the angle θ and the distance between the points. The PFH parameters are the dimensions considered to compute the histogram and the number of bins for each dimension.

Histogram of oriented gradients

The *histogram of oriented gradients* (or HOG) is computed from a set of gradient orientations extracted on a dense grid in the image patch. It is a regional descriptor composed of a set of small local histograms of gradient orientations computed on the grid. The main idea of HOG is that local appearance and shape can often be characterized by the distribution of local intensity gradients. Figure 2.7(a) shows gradient orientation histograms for the window corner image patch. A histogram is computed for each cell in the grid and is plotted as a rose. The HOG operator also includes a dependency on image location of the local histograms. Although not rotationally invariant (as SIFT), it is normalized with respect to image contrast and, therefore, its biggest advantage is that it is robust to changes in brightness. The parameters of the HOG are the grid size and the number of orientation bins.

The Gist Descriptor

The Gist encodes the "shape of a scene" as a unitary object. It is a holistic modeling of a scene that encodes a set of perceptual dimensions such as naturalness, openness, roughness, expansion, ruggedness, which, together, represent the dominant spatial structure of a scene. The Gist descriptor estimates these properties using spectral and coarsely localized information. This is extracted by means of discrete Fourier transforms which capture both the unlocalized spectral information and its spatial distribution. Thus, the Gist encodes the global configuration of the image, ignoring most of the details and semantic object information. The descriptor showed good results for scene classification, when combined with color information [Quattoni and Torralba, 2009]. Figure 2.8 shows the visualization of the gist descriptor characterizing a bar scene.

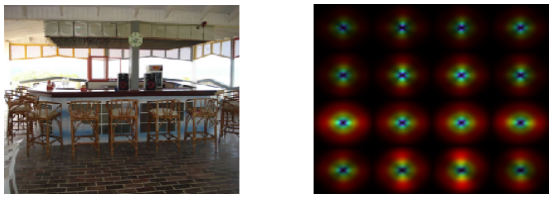


Figure 2.8: The GIST descriptor of a bar scene.

Bag of Words and Spatial Pyramids

The *bag of words* (or BoW), also denoted bag of features, is an image representation method that includes the following steps: feature detection, feature description and codebook generation. Feature descriptors characterize interest points and they are further used to obtain the codewords, which are representatives of several similar patches, characterized by similar feature descriptors. The dictionary of codewords is obtained by clustering all descriptor vectors. Codewords are then defined as the centers of the learned clusters. It follows that each patch in the image can be mapped to a certain codeword through the clustering process. As a result, the image can be represented by a histogram of codewords. Starting from the histogram, one can learn a BoW model (e.g., a discriminative kernel-based classifier) which can be further employed for recognition. A limitation of BoW is that it completely ignores spatial information. Several methods were proposed to upgrade the BoW representation in this direction.

A popular one, which we also use in this work, is *spatial pyramids* (or SPs) [Lazebnik et al., 2006]. They partition the image into increasingly fine sub-regions and then concatenate histograms of codewords (or features) found inside each sub-region. The sub-regions are obtained using a three-level grid of granularity. The first level is the image itself (thus, the BoW histogram), the second level is a 2x2 grid and the third level is a 4x4 grid. A histogram of words is computed in each block of the grid (or sub-region) and then concatenated into one feature descriptor. Figure 2.9 is a schematic illustration[4] of the spatial pyramid representation for an image.

[4]This is for demonstration purposes only, and it is not an exact calculation of the feature descriptors for the exemplified image.

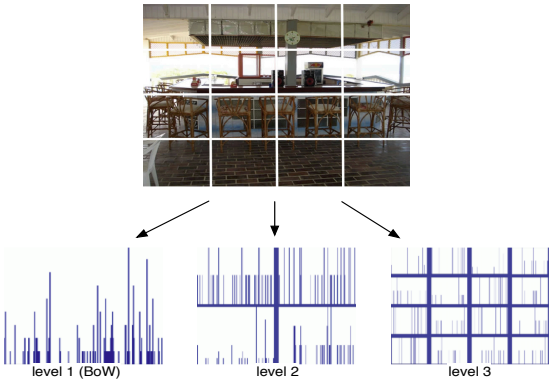


Figure 2.9: An illustration of the BoW and the spatial pyramid representations.

2.2.3 Object Recognition and Scene Understanding

Scene understanding, also named image understanding, as introduced in Chapter 1, is a complex problem which establishes correspondences between an image and a symbolic knowledge base. It covers the task of object recognition via the notion of *context* in which objects occur. In many circumstances, object recognition heavily depends on the surrounding objects and scene elements. From this point of view we can distinguish three broad axes of methods that attempt to solve scene understanding problems, which build on concepts that use simpler or more complex information in the image. Methods that consider objects in isolation and do not explicitly consider their structure are denoted as *appearance* methods. These are based on low and mid-level features, such as the ones introduced in the previous subsections. Appearance-based methods ignore the geometric and structural information about the key parts in the image, and usually fail when faced with variable part configurations and occlusion. *Contextual* methods make use of contextual information outside the object itself (objects are not treated in isolation), however, they do not necessarily consider a structural representation of the problem. Finally, *structural* methods represent visual information in a structural form (e.g., by employing graphical models, grammars).

This view is also related to the learning method employed. For appearance-based methods it is more likely that discriminative, and thus, propositional statistical learning techniques are employed (e.g., template matching, boosting, SVMs). Structural methods are often employed with generative learning, and thus, graphical models and grammars (e.g., fixed compositional structures [Felzenszwalb et al., 2010], constellation models [Fergus et al., 2007]). Contextual

methods are distributed among the two main learning techniques and, in this case, the difference comes more from the use of spatial information and the co-occurrence of other elements in the image.

Appearance-based techniques include *template matching*, which we use as a baseline in our work. The basic idea of template matching is to use a convolution mask (or template), adjusted to a specific feature of the search image, which we want to detect. On a test image, the convolution output will be highest at places where the image structure matches the mask structure. To be able to somewhat deal with inter-class variations, in practice, a deformable template is used [Torralba et al., 2004]. *Part-based* models are examples of structural approaches that recognize an object by finding its constituent parts and measuring their geometric relationships. They can have different topologies for the geometric connections and more complex hierarchical part-based models can be derived towards a grammar [Girshick et al., 2011]. We will give a more detailed overview of the related work in the next chapter, where we present historical insights and latest structured approaches to scene understanding.

2.2.4 Robot Grasping

Given an object, the goal of robot grasping is to find a *grasp hypothesis*, or a relationship between an object and a robot hand/gripper, that allows for some subsequent manipulation of the object. Robot grasping includes several steps. A first step is to perform a *reconstruction of the environment* and to segment and recognize the object. Once the object is detected and segmented, the robot needs to estimate its pose (position and orientation) in order to proceed to the *grasping strategy* step. Often, before the approach strategy is planned – that is the arm motion and hand configuration are planned for grasping, there is a *pre-grasp* step for object adjustment [Kappler et al., 2010]. Such pre-grasp decisions or manipulation actions bring objects into better configurations for grasping. Examples of pre-grasp decisions are object rotation, object sliding or, if the object is not within (the best) reach, arm moving to detect the best pre-grasp pose, while avoiding obstacles. The final step is *grasp execution*: placing the fingers on the object and making sure that the goal was achieved.

The role of a grasping strategy (or grasp synthesis/hypothesis) is to find a grasp configuration that satisfies a set of criteria which are relevant to accomplish a grasping task. This implies achieving grasp stability, task compatibility, adaptability to novel objects and other grasping constraints. Figure 2.10 shows the four main sources of influence on a grasp hypothesis selection that a grasping strategy must consider: *gripper parameters*, *object properties*, *environment constraints* and *task constraints*. Also, because of the variety of object shapes

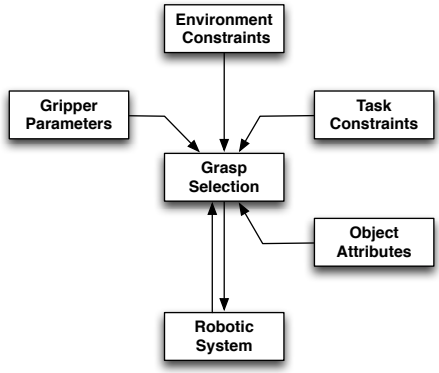


Figure 2.10: Strategy of grasp selection.

and sizes, a grasping strategy should be good enough and prepared to grasp novel objects. Thus, selecting a suitable grasp among the infinite set of candidates is a challenging problem in robotics.

A first important role in grasping is played by the gripper. The function of a gripper is to grasp objects and possibly manipulate them with its fingers. During task execution, the grasping fingers must be controlled such that the grasp is characterized by dexterity, stability and dynamic behavior. This requires methods of computing the gripper parameters, such as position, orientation and forces of fingertips and joints. Grasping stability often relies on the local geometry and on the limitations in the contact force transmissions. A grasp is stable if a small disturbance, on the object position or finger force, generates a restoring wrench that tends to bring the system back to its original configuration. All these aspects imply a large number of parameters. A big challenge is finding ways to map the parameters to simple and general representations, such that searching for good grasps is fast.

Another important role for a successful grasp is played by the object and its properties. Grasping approaches can be divided based on whether they address grasps for known or unknown objects. In the case of known objects, most approaches are analytical and based on object recognition and pose estimation [Zhu and Ding, 2004, Prats et al., 2007]. They determine the contact locations on the object and the hand configuration that satisfy task requirements through kinematic and dynamic formulations. Grasp synthesis is then usually formulated as a constrained optimization problem over criteria that measure gripper parameters. In this case, a grasp is typically defined by the grasp map that transforms the forces exerted at a set of contact points to the object.

Analytical approaches are good to find stable, but not task-oriented grasps. Task-oriented analytical approaches suffer from computational complexity as they fail to find a general mathematical formulation compatible with different tasks.

In the case of unknown objects, most approaches are empirical and based on learning methods [El-Khoury and Sahbani, 2010, Lenz et al., 2013, Song et al., 2010]. They rely on sampling grasp hypotheses for an object and are further divided into: techniques centered on the observation of a human performing the grasp and those focused on the properties of the grasped object. In the first case, a robotic system observes a teacher performing a task and it tries then to reproduce the same grasps. Such techniques are known as learning from demonstration or *human centered* [Kjellström et al., 2008]. They overcome the difficulty of analytical approaches by learning the task. However, these systems are not fully autonomous when they face new objects or new tasks. To overcome this problem, *object centered* methods [Saxena et al., 2008b] learn grasp - object feature mappings. They are capable of generalizing to new objects as they are more adaptable via learning. They learn to find good grasping poses or to associate object local features to different hand parameters. Their drawback is, however, that they generate too many possible grasping hypotheses. Therefore, finding a task-compatible grasp for a new object is still an open challenge. In Chapter 6 we propose a new solution for task-dependent robot grasping based on probabilistic logical reasoning. In addition, we propose a new SRL approach to pre-grasp position recognition which can generalize to different object categories, based on contextual object shape.

Part I

Relational Scene Understanding

Chapter 3

History of Relational Representations in Visual Recognition

This chapter provides insights in the history of syntactic, relational and statistical pattern recognition for visual recognition problems. The goal is to review the use of high-level relational representations for visual recognition built on symbolic descriptions of the image content and point out ideas that were central in knowledge-based early vision work. We emphasize intuitions that have been forgotten or are not enough appreciated by the vision community today. For reference, we rely on four ingredients: data, representation, learning and world knowledge. Critical for modern computer vision and visual recognition solutions, they are also dependent on each other and interrelated. With this in mind, we structure this chapter along three, roughly delimitated periods.

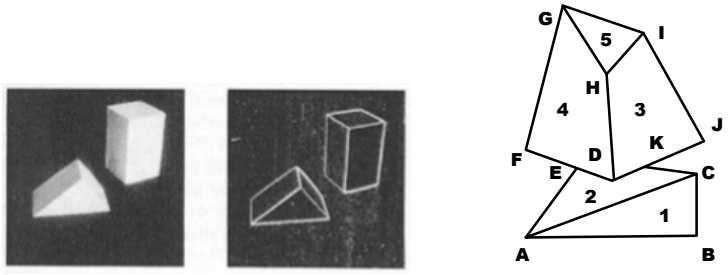
The first period is that of *syntactic pattern recognition* (Section 3.1). It marked the early vision era between 1960s and 1980s. A pattern is a quantitative or structural description of a concept or some entity of interest. While decision-theoretic pattern recognition is based on decision functions and is suitable for applications where patterns can be meaningfully represented in vector form, syntactic pattern recognition has structure-handling capability and is suitable for domains where structure plays an important role, e.g., scene analysis [González and Thomason, 1978]. Following the idea of decomposition of patterns into subpatterns and/or primitives, early visual syntactic pattern recognition was concerned with reasoning about scenes and finding the most likely object

interpretation for regions in an image. In this period, the recognition problem was dominated by the discovery of analytic and relational representations of objects and often relied on geometric properties. The underlying motivation of these descriptions was to perform recognition under partial occlusion, any viewpoint and any condition of illumination.

Section 3.2 continues with take away messages and lessons learned from early vision. It indicates why large scale visual recognition was (and still is) a hard problem and why older ideas trying to incorporate symbolic and relational knowledge were abandoned. The high-level vision progressed at that time by often assuming (prematurely) that the low-level vision part was solved. Insights that this assumption was flawed led to the second period, that of *statistical pattern recognition*. Motivated to overcome the shortcomings of low-level routines and encouraged by the strong launch of statistical machine learning, computer vision moved firmly in 1990s towards statistical pattern recognition. Since then and until today, low-level features have been a major research thrust in the vision community.

Although there was a dominance of statistical or syntactic pattern recognition for visual tasks in different time periods, recognition ideas were continually re-visited, as computational power, feature construction, reasoning/learning methods, and data availability advanced. The two areas pursued their goals somewhat independently and in parallel and were not totally abandoned at any point in time. For instance, in the early decades, ideas from signal processing, such as autocorrelation, were exploited for character recognition. Similarly, symbolic and relational representations re-appeared not long ago and revisiting them is currently of growing interest. Section 3.3 covers the last period which is marked by recent trends that combine statistical machine learning with symbolic and relational world knowledge. We indicate what is different now and we overview current SRL work in visual recognition.

In Section 3.4 we conclude with potential gains for the computer vision community if SRL succeeds, by pointing out what it can add to solve current and older problems. Along the exposure we point out what were the successes and weaknesses of each approach, while focusing on the visual recognition problem. We present the advances from the relational/structural representation point of view. Other surveys on early vision work can be found in [Mundy, 2006, Nilsson, 2009], however they are less focused on relational representations.



(a) (b)

Figure 3.1: Blocks world scenes; (a) illustrates the blocks world in [Roberts, 1963] (photograph used with permission of Lawrence Roberts), while (b) shows the representation scheme of vertices, regions and edges used by [Guzmán, 1968].

3.1 Back in History: Syntactic Pattern Recognition

The early goal of computer vision, started at MIT, was to develop a set of routines based on formal logic and other mathematical tools that could analyse a picture. It started with blocks world, a simplification of the world, where objects were restricted to polyhedral shapes on a uniform background and in an arbitrary spatial arrangement that could occlude each other. The goal was to name shapes projected in 2D with symbols from a predefined vocabulary (see Figure 3.1(a)).

The blocks world domain led to several early vision systems that used reasoning to interpret visual scenes. The first system was that of Roberts [Roberts, 1963]. His recognition algorithm included a library of polyhedral parts that could be assembled in different configurations to obtain composite structures. His idea was that *"objects could be constructed out of parts with which we are familiar. That is, either the whole object is a transformation of a preconceived model, or else it can be broken into parts that are known"*. Roberts' algorithm was based on grouping heuristics modeling constraints of polyhedral scenes.

Early Image Description Languages Guzmán's system SEE [Guzmán, 1968] continued the recognition work in the blocks-world domain. Its task was to segment scenes into plausible three-dimensional constituents called "bodies". The input scene was represented in a symbolic format, in terms of straight lines, plane surfaces (or regions) and vertices (line intersections), as Figure 3.1(b) illustrates. The recognition problem was formulated using a figure description language which could naturally represent configurations of regions and lines together with their properties, symbolic relations, such as "strong" or "weak",

and composite relations, such as "nucleus" identifying a region or a set of nuclei. The language was implemented in the Lisp language and the relations were defined as background knowledge using rules. For example, the definition of a "nucleus" was: "if two nuclei are connected by two or more links, they are merged into a larger nucleus by concatenation". SEE could solve quite complex polyhedra configurations. However, there were many drawbacks of the system. It was not able to identify shape symbols such as "cubes" or "houses", leaving this task to a higher-level module. Further, the experiments consisted in several perfectly painted scenes or pictures of painted blocks. Thus, no noise and shadows were considered and a few mistakes were enough to fool the program.

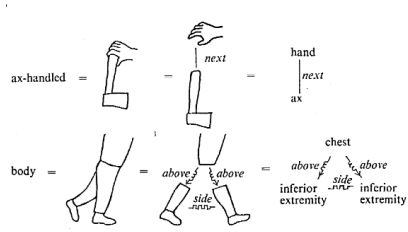
Similar local and global constraints for polyhedral scenes were exploited and improved later [Huffman, 1971, Clowes, 1971, Waltz, 1975]. In these works, the local constraints of vertices and edges were put in context and global inconsistencies were checked to rule out multiple scene interpretations. The blocks world domain dominated vision research for over a decade before it was largely abandoned in favour of more realistic scenes. This did not imply, however, that the problem of recognizing polyhedral objects and more complex structures using polyhedra was completely solved, even when solutions assumed other simplifications (e.g., that primitive lines were given). Several papers continued working on syntactic pattern recognition for visual recognition until late 1980s (e.g., [Ferraté et al., 1988]). Meanwhile, the community realised that even if the proposed approaches solved the blocks world problem, they were not likely to hold in real-world scenes. This interplay between formal theoretical frameworks and the ability to apply them on complex real-world problems is valid for computer vision research field even today.

Graphs A first step in moving away from the blocks world towards more realistic domains was made by Guzmán. He extended the blocks world with curved objects [Guzmán, 1971]. The central concept was that of "relation models". They represented a scene as a symbolic graph between regions (with relations such as "next to" and "above" to name a few). In addition to the relational aspect both local and global constraints were used. Thus, context was considered an important element of the system. The relational models had the form of "parsing trees" (see Figure 3.2(a)). However, besides the fact that there was no implementation of the proposed approach, there were two other main drawbacks. One was the perfectly assumed segmentation problem. The other one was the assumption of ideal primitive lines which was too far away from real-world applications. As a result, the move was made towards the generalized cylinder domain, which extended the blocks world with composite curved shapes [Nevatia and Binford, 1977].

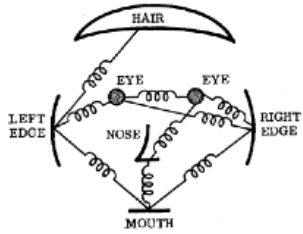
Guzmán's trees were not the only graph-based representation used in early vision. Another representational scheme was "pictorial structures". A pictorial structure

was a reference model composed of a number of rigid components held together by "springs" or linkages [Fischler and Elschlager, 1973]. The rigid piece could be a small image patch or the entire reference image and corresponded to a single coherent entity in the image. An example is illustrated in Figure 3.2(b). The springs joining the components were serving as relative displacement constraints among the rigid pieces with costs of the displacements. The springs costs captured semantic information (which was application dependent) and syntactic information (which consisted of the number of components, the embedding metric, etc). The problem of matching a pictorial structure to an image was defined as an energy function to be minimized in order to find the best match. The cost of a particular configuration depended on how well each part matched the image data at its location and how well the relative locations of the parts agreed with the model. Some qualitative experiments on a variety of line type drawings and grey-level images were presented in [Fischler and Elschlager, 1973].

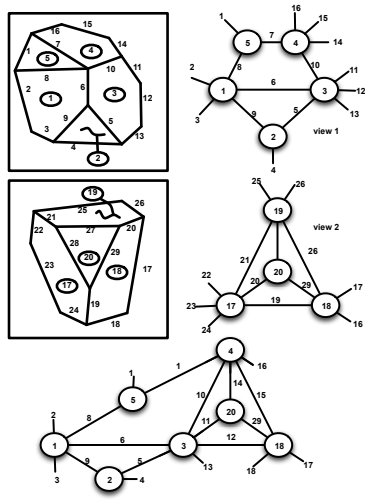
Further, “aspect graphs” were also used as a representational strategy. The idea behind aspect graphs is to represent a polyhedral shape using multiple related 2D views of the object. An aspect graph is a graph with a node for every aspect (or object view). An aspect is the topological structure of the object from a viewing direction. Graph edges connect adjacent aspects. They arise from the topological structures of the views, i.e., transitions in the graph structure relating surfaces and edges of the projected object. Figure 3.2(c) shows an instance of an aspect graph extracted from two object views. Aspect graphs were promoted by [Underwood and Coates, 1975] in the form of interpretation trees. An interpretation tree consists of different object descriptions which can be produced from the object views. The tree shows various ways in which the object views can be matched to the previously generated description. An interpretation tree in graphical form is shown in Figure 3.2(e). The correct description is the one with the smallest matching error value. Aspect graphs were also extended to generalized cylinders [Ponce and Kriegman, 1990]. They encountered major difficulties later because of the graph size explosion due to scale transitions that were relevant topologically, but unimportant from the recognition point of view. Another characteristic of aspect graph-based approaches is that they were model-based. Working top-down, they tried to fit relational models to meaningful ungrouped features. Such works included recognition plans by [Goad, 1983] and interpretation trees by [Ikeuchi, 1987, Grimson and Lozano-Pérez, 1987]. Differently, in bottom-up approaches first low-level features are extracted and then they are grouped according to defined patterns. However, given the low-level vision performance at that time, bottom-up approaches showed little stability and did not scale to complex textured objects.



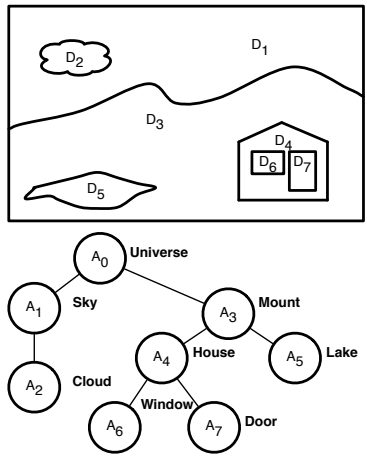
(a) Relational models in the form of parsing trees by [Guzmán, 1971] (illustration used with permission of Adolfo Guzmán).



(b) Pictorial structure representing a face introduced by [Fischler and Elschlager, 1973] (illustration used with permission of Martin Fischler).

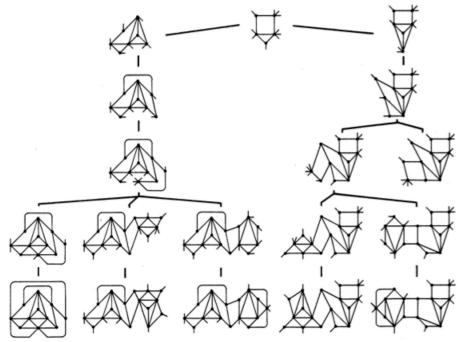


(c) Aspect graph extracted from two views (excerpt reconstructed from [Underwood and Coates, 1975]).



(d) Semantic map representing the scene above it used by [Preparata and Ray, 1972].

Figure 3.2: Examples of graph-based representations used in early vision.



(e) Image interpretation tree. Each node shows a different way in which object views can be matched. The image was taken from [Underwood and Coates, 1975] under copyright notice © 1975 IEEE.

Figure 3.2: Examples of graph-based representations used in early vision.

In [Barrow and Popplestone, 1971, Preparata and Ray, 1972] relations between regions and their properties were stored in the form of a graph or semantic map; the nodes corresponded to the regions in the image (or surfaces/ parts of an object), while edges to relations between objects. Interpreting a scene consisted of graph or subgraph matching operations. A similar relational matching approach based on graph isomorphism was proposed by [Ullmann, 1983] and [Shapiro and Haralick, 1982]. The matching approach was able to handle n-ary relational structures in which n-tuples (or symbolic edges) represented complex relations between primitive image parts. The motivation behind the idea of matching relational representations instead of whole pictures was that one can make the matching process invariant under changes of appearance attributes that one can ignore. Additionally, relationships could be used contextually to recognize primitive parts. A major drawback was the high computational cost of matching high arity relations with many objects in the database. A solution to this problem proposed then was to do the matching only against scene prototypes obtained by clustering object descriptions. It was implemented by the VPI vision system [Shapiro, 1983]. However, the performance of the low-level visual routines at that time restrained empirical evaluation. Another weakness of the approach was dealing with occlusions or rotations of objects which changed the graph structure dramatically.

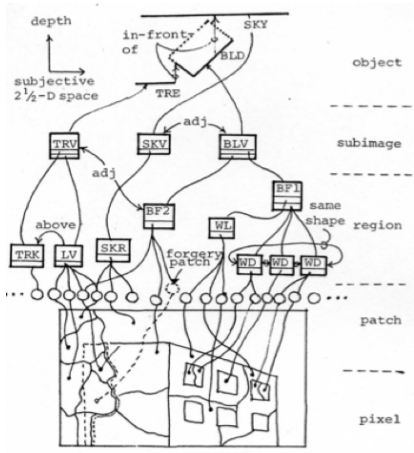
Relational Compositional Hierarchies Several other (hierarchical) logical and relational representational schemes for model-based image understanding were proposed in more realistic domains. Some early approaches based on

relational description languages are surveyed in [Kanade, 1977]. Starting from five levels of image description, i.e., pixel, patch (or line segment), region, object (part of the image corresponding to an object) and semantic object (the object itself) layers, Kanade indicates different forms of dependencies between the elements. We discuss the ones that are based on relational descriptions. Figure 3.2(f) illustrates qualitative relations at the object level.

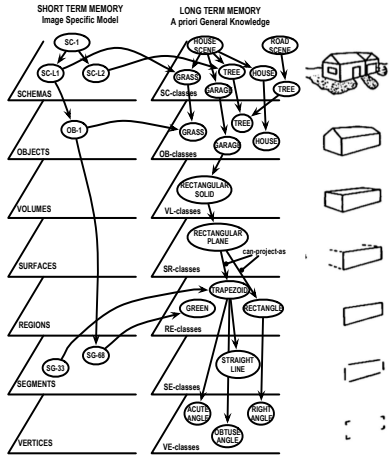
In his survey, Kanade stresses the role of the hierarchy and the control in linking its layers, but also the importance of a proper image representation and a good integration of semantic constraints at different layers in the hierarchy, when dealing with noise. First steps into this direction were taken by [Ohta et al., 1979] and [Ohta, 1985]. These works combined bottom-up processing with top-down control for semantic segmentation of outdoor images. Starting with an over-segmentation, the system generated "plan images" or scene interpretations by merging low-level segments. Domain knowledge was represented as a semantic description map in the bottom-up process and a set of production rules in the top-down process.

Another relational hierarchy was proposed in [Hanson and Riseman, 1978] (Figure 3.2(g)). Following the authors' position that the "*interpretation of an image involves the construction of an internal model which is a description of the major semantic elements in the scene, as well as their three-dimensional relationships in the physical world*", the work proposed an approach that combined the low-level result of visual segmentation with the high-level interpretation of a scene. The system included a relational, explicit representation of knowledge to describe the world, the construction of an appropriate model for the task at hand and the control of the matching process. The interpretation of a particular image was defined as an inter-linked collection of instantiations of concepts of interest. The set of instantiations was named "short-term memory" and general a priori knowledge was called "long-term memory". In relational learning terminology they are equivalent to the set of facts and background knowledge, respectively. The knowledge had the form of directed semantic networks, called "schemas". Schema instances communicated with each other via a global blackboard, which have advantages for parallel implementations. The knowledge representation was, however, not mainly declarative, because of the recognition control strategy used. Although the resulting system VISIONS was general and could, in principle, be applied to a variety of domains, the empirical evaluation was limited. It consisted of qualitative results for one image and quantitative results for two images.

Logical and Relational Languages Starting from the late 1970s more formal standpoints arose tackling semantic and syntactic visual recognition. As pointed out in [González and Thomason, 1978], "*structure and relationships of the various components of a scene are of fundamental importance in establishing*



(f) The compositional hierarchy from [Kanade, 1977] (illustration used with permission of Takeo Kanade).



(g) The hierarchical relational framework implemented by the VISIONS system and proposed in [Hanson and Riseman, 1978] (illustration reconstructed from the original).

Figure 3.2: Two hierarchical logical/relational representational schemes for model-based image understanding in early vision.

a meaningful recognition scheme". Based on concepts from formal language theory, formal grammars and languages have been proposed to handle complex representations such as trees or graphs. They include logical formalizations, such as first-order logic, to interpret scenes. Such frameworks were employed in several systems in the late 1980s. A presentation of this work can be found in [Peraldi et al., 2011].

The work of [Yakimovsky and Feldman, 1973] and [Tenenbaum and Barrow, 1977], tried to solve the segmentation and region recognition problems starting from detected patches. They used semantic constraints such as "adjacent" and "above". Different optimization and filtering procedures were used to find the most likely explanation of an image. [Yakimovsky and Feldman, 1973] developed a Bayesian framework for analyzing road scenes. It combined segmentation with semantic domain information at the region and inter-region level. [Tenenbaum and Barrow, 1977] proposed an interpretation-guided segmentation which used constraint propagation to get a globally consistent scene interpretation for the region labels. Although this was quite impressive at that time because of the method and representation used, the empirical results were not very good. The main problems were the weak performance of the lower-level visual routines

which could have helped to guide the segmentation better.

A relational and declarative approach to represent and use high-level knowledge for object recognition was proposed in [Silberberg, 1987]. Objects were modeled using collections of regions and lines, taking into account spatial, temporal, and contextual knowledge. A semantic network was used to represent relationships between objects. The recognition process was both bottom-up and top-down, following the hypothesize-verify scheme. However, the reasoning part suffered from computational problems when many objects were involved in the scene.

The logic-based approaches proposed were built on declarative representation languages with a formal semantics. The representation relied on initial symbolic descriptions extracted from images. A first logic-based formal theory was employed in [Reiter and Mackworth, 1987, Reiter and Mackworth, 1989] for hand-drawn sketches of geographical regions. Their interpretation system used image-domain knowledge, corresponding to the concrete object instantiations found in the image, scene-domain knowledge, corresponding to general domain knowledge, and a mapping between the image and scene domains using first-order logic. An interpretation of the image was a logical model of a set of logical facts or formulae, which described domain knowledge as well as low-level processing output. Example 11 illustrates first-order logic axioms expressing knowledge about the taxonomy of image-domain objects. The goal was to compute an interpretation of the image in terms of roads, rivers, pieces of land and water. To make the computation of all possible interpretations or models feasible, Reiter and Mackworth considered the close world assumption, which allows reducing the first-order formulas to propositional formulas. They use scene consistency to prune interpretations that are impossible. The computation of all possible models is formalized as a constraint satisfaction problem. The approach assumes deterministic image descriptions, neglecting noise and incompleteness aspects. This implies too rigid and strict assumptions for a general purpose scene interpretation framework. In [Poole, 1993] a similar interpretation scenario was exemplified in a probabilistic knowledge setup, however no experimental results were provided.

***Example* 11.** *Image-domain knowledge specified as first-order logic axioms in [Reiter and Mackworth, 1989]:*

$$\forall x : image\_object(x) \Leftrightarrow chain(x) \vee region(x)$$
$$\forall x : \neg(chain(x) \wedge region(x))$$

The first axiom states that chains and regions are the image primitive objects that can exist in a geographical map, whereas the latter axiom states that an object cannot be both chain and region at the same time. Relations between image-domain chains are specified using predicates such as $tee(o, o')$*, meaning that chain* o *meets chain* o' *at a T-junction.*

Differently, the first-order logic approach in [Matsuyama and Hwang, 1985] does not assume the availability of an a priori image segmentation and constant symbols representing image-domain objects. These are created through an expectation-driven segmentation reasoning and are the output of the interpretation process. They are computed through hypotheses generation starting from observations (or evidence) and reasoning backwards to find feasible explanations (or hypotheses). An advantage of this approach is that it can deal with situations where the available information is incomplete. The hypotheses generation computes interpretation networks consisting of mutually related object instances. The reasoning system returns the set of hypotheses that are possible. The language models classes of scene objects, their attributes and spatial relations using logical rules, functions and logical constraints. The approach was applied to aerial images of suburban areas that show houses and roads. Example 12 illustrates first-order logic axioms that are used to represent general knowledge about this application domain.

***Example* 12.** *The* SIGMA *system represents the fact that every house is related to exactly one street as follows:*

$$\forall x : house(x) \Rightarrow \\ (\exists y : road(y) \wedge related(x, y) \wedge \forall z : (road(z) \wedge related(x, z)) \Rightarrow z = y)$$

Grammars Other formal frameworks tackling computational theoretical aspects of visual recognition were automata and grammars. Popular formalisms in early vision were tree, shape and graph grammars [González and Thomason, 1978]. The recognition of syntactic structures with these formalizations was performed using different automata.

An experiment on fingerprints using deterministic tree grammars is present in [Moayer and Fu, 1976]. The experiment consisted in 92 image samples, from which 193 patterns were parsed and recognized by the system. Many images, however, were too noisy to be handled by the grammar. Thus, there were several drawbacks of these formalisms when applied on real problems. One of them was efficiency when parsing. Another disadvantage was handling the noise, as only patterns without any error were accepted by the grammar. Different correction steps were employed to tackle this problem. The other alternative was to incorporate probabilities in grammatical formulations, however, no real experimental results were obtained until much later [Zhu and Mumford, 2006].

In turn, the work by [Rosenfeld, 1979] starts from fundamental questions, such as what types of recognition tasks are feasible and what are the minimum operations required to perform a task, and studies various types of automata defined for two-dimensional arrays that can recognise certain classes of pictures (including highly parallel automata and stacked automata). Then, Rosenfeld

links at a theoretical level his proposed array automata for bottom-up recognition to more general (parallel) array grammars, which can be used both for top-down and bottom-up image parsing. Similar in spirit, Fu presents in his book [Fu, 1982] several applications that employed grammars for visual recognition. One of them is shape analysis of contours, for which an hierarchical syntax shape analyzer was developed in [Pavlidis and Ali, 1979]. However the focus of the book is on attributed grammars, which were introduced for shape recognition in [You and Fu, 1979]. Although not probabilistic, they showed robustness to noise due to the extra contextual rules imposed on the attributes. Nevertheless, in a noisier environment a correction step was still required.

Early Relational Systems The relational, logical and hierarchical formalisms presented produced several pioneering systems. One of them was MSYS [Tenenbaum et al., 1975]. It was able to reason about and find the most likely interpretation for the regions in the image scene, given a number of interpretations and their probabilities. As stated by the authors, "MSYS is a system for reasoning with uncertain information and inexact rules of inference". Another example is the system VISIONS [Hanson and Riseman, 1978]. It was an ambitious system that analyzed a scene on many interrelated levels including segments, 3D surfaces and volumes, objects, and scene categories. In addition, it had integrated contextual knowledge, such as 'road scenes', which were used to group contextually related objects. However, their main drawback was to assume that region primitives were given a priori, and therefore, manually segmented. Similar declarative and logic-based representations, utilization of domain knowledge and control structures were employed by Mapsee [Reiter and Mackworth, 1989] for geographical region interpretation and by SIGMA for aerial image understanding [Matsuyama and Hwang, 1985]. Because the application involved less difficult low-level processing, the proposed frameworks showed that the approach was feasible in practice if the low-level part was solved. Other promising results in this direction were obtained by the systems ACRONYM [Brooks, 1981], VPI [Shapiro, 1983] and SPAM [McKeown et al., 1985]. They used low-level and intermediate image processing to get the image primitives and world knowledge to reason about scene interpretations. One decade later, shape grammars were employed in a crisp logical form by the system FORMS [Zhu and Yuille, 1996].

It is fair to say that early vision work and systems were clearly dominated by a knowledge-directed paradigm. For this purpose, often large amounts of specific knowledge were collected about the objects and domains. This resulted in quite complex object class descriptions which were integrated by the vision systems, making the reasoning systems quite complex and sometimes difficult to be applied in practice.

3.2 Lessons Learned and the Move towards Statistical Learning

Obtaining general vision systems was even more difficult in the early vision era than it is today. As already pointed out, the (relational) approaches and systems in those times had several major drawbacks. One of them was the simplification of vision recognition applications to blocks world, generalized cylinders and some extensions, which were far away from real-world problems. Even so, the blocks world and generalized cylinder systems were heavily dependent on the rigid and strict models and had difficulty dealing with noise. That was due to the fact that these scene interpretation systems using spatial constraints on the image parts relied on good low-level processing which ensured the primitives. In case of a poor segmentation, however, they could fail completely. This was often the case, as the noise-prone low-level features, even in simple domains such as the blocks world, were immature and little advancement had been shown until early 1980s. This led to several systems which were only working in a small domain world. Therefore, there was a growing realization [Shapiro, 1983] that in order to have intelligent and general vision systems, they must show robustness to noise, occlusion, illumination, scale, and so on. Linda Shapiro notes:

> *"We have showed the use of relational representations, we must yet discover the use of low-level knowledge, not about specific objects such as airplanes or chairs, but about all wooden objects with curved surfaces or all metal objects with flat tops." Linda Shapiro, 1983*

Furthermore, the shortage of data was an important obstacle. However, even if data was available, another major drawback was the lack of learning techniques to make the systems more robust to noise and able to better generalize to new data. As the application domains broadened, the available systems (i.e., mainly large systems with a single reasoning control engine) became too complex and difficult to be applied. Directly linked to this matter, another shortcoming was the fact that the data representation at all processing levels and the interpretation procedures were mixed in a single engine. Denoted as the *control process* by [Draper et al., 1996], the recognition procedure itself (i.e., matching or reasoning) was never addressed as an independent problem by the early vision work. Draper et al. note:

> *"In particular, we argue that while these systems addressed (...) many critical issues, their success was limited not only by the relative immaturity of the field, but also by fundamental and still open*

> *problems in control. (...) The knowledge engineering paradigm used to collect knowledge for small systems was inadequate for gathering the larger amounts of knowledge needed for more general systems. In addition, as the size of the knowledge base grew, the systems integration problems became more and more dauting." Draper et al., 1996*

Much of the high-level knowledge about the structure of the scene was not clearly separated from the routines implementing the low-level processing. This drawback was already pointed out by Kanade 20 years earlier in his survey [Kanade, 1977] and it was also noted by Russ et al. in [Russ et al., 1996]:

> *"We believe that all of these systems suffered from the major drawback that they forced one to commit oneself to developing large vision systems with a single inference engine, whereas there were and are many tasks in vision better handled by different control structures." Russ et al., 1998*

Finally, the computational power was another important impediment. Derek Hoiem notes in his paper [Hoiem et al., 2007]:

> *"It is interesting to note that a lot of what are considered modern ideas in computer vision—region, superpixels, combining bottom-up and top-down processing (...) were well-known three decades ago! But, though much was learned in the development of these early systems, none of them were particularly successful, mainly because of the heavy use of hand-tuned heuristics which did not generalize well to new data. This, in turn, led people to doubt the very goal of complete image understanding. However, it seems that the early pioneers were simply ahead of their time. They had no choice but to rely on heuristics because they lacked the large amounts of data and the computational resources to learn the relationships governing the structure of our visual world." Derek Hoiem, 2007*

These limitations marked the end of an active period for relational and structural methods and the rise of appearance methods and statistical approaches, once machine learning started to get momentum. However, syntactic and structural visual recognition was not totally abandoned at any time. It was kept alive in the next two decades by few dedicated researchers [Bunke and Sanfeliu, 1990, Esposito et al., 1992, Russ et al., 1996]:

> *"This book is currently the only one on this subject [syntactic and structural pattern recognition] containing both introductory material and advanced recent research results. It presents, at one end, fundamental concepts and notations developed in syntactic and structural pattern recognition and at the other, reports on the current state of the art with respect to both methodology and applications."*
> *Bunke and Sanfeliu, 1990*

Following different goals and problems (i.e., improving low-level vision routines), the interest of the computer vision community was shifted to statistical pattern recognition. One example is the VISOR system which employed neural networks for scene analysis [Leow and Miikkulainen, 1994].

3.3 Bringing Back Relations in Visual Recognition

After more than two decades of using purely statistical methods in computer vision, large scale vision recognition remains a hard problem [Fei-Fei, 2013]. Nevertheless, given the burst of computational power, data and statistical learning techniques, impressive progress in low- and mid-level vision has been made. Derek Hoiem notes in 2007 that: *"the advancement of learning methods in the last decade brings renewed hope for a complete scene understanding solution"*. Although statistical techniques using solely appearance cues in the form of low and mid-level features are able to solve many real-world problems, they are insufficient to obtain complete scene descriptions when high-level tasks and structured domains are involved. Therefore, it may be the right time to build on top of them and bring back in visual recognition the early ideas that hierarchical structure and symbolic relations are key components of an image understanding system.

This section is meant to put face to face old and new approaches for relational visual recognition and inspect how they evolved in time. The goal is to show that along with the progress to overcome early vision problems, old ideas have started to be re-investigated. The recent trend in visual recognition is to move again from purely statistical methods to methods that combine syntactical/structural and statistical aspects in order to take the best of both worlds.

3.3.1 Timeline and Axes for Discussion

The axes for discussion are generated from Table 3.3.1. Resulting cells cover a specific representation of the problem (given by a row) and the way this

representation is used, that is the decision making technique (given by a column). We consider as representational formalisms graphs, logic and relational languages, compositional hierarchies and grammars. We group the decision making aspect along three main lines: deterministic matching/reasoning/learning, graphical models and remaining statistical methods (both parametric, i.e., kernel methods, linear classifiers, neural networks; and nonparametric, i.e., KNN, decision trees, random forests). Among the statistical decision making approaches, graphical models are listed separately as they are a popular method in the computer vision community. Therefore, when exposing the related work, we merge the last two columns for all types of representation, except graphs. In addition, since there is relevant recent work in employing deterministic relational approaches for visual recognition, we will allocate to it an extra axis for discussion from a modern perspective. The rest of the cells, identifying representation – deterministic techniques combinations, were already presented in Section 3.1. Results axes of related work are: relational/logical matching, learning and reasoning, graphical models, other graph-based statistical approaches, statistical relational learning, stochastic grammars and probabilistic compositional hierarchies.

To indicate the trends in visual recognition in time or how old ideas are being revisited, we fill each cell with old and new references to papers that belong to the specific area of discussion. For practical reasons, the table covers part of the literature references. We picked the ones that are more representative especially for the recent related work. We also point out in the text the advancements of the recent work as opposed to the old one. This setup gives us a playing ground to mark the potential direction of SRL in visual recognition.

3.3.2 Recent SRL-related Work in Visual Recognition

The axes of related work are described as follows.

Graphical models Visual recognition tasks heavily rely on appearance and context information from objects. Appearance information is based on visual cues and can recognize categories up to a certain degree. Context information, based on the interactions between pixels, object parts or objects in the scene, or on global cues, can help further to disambiguate appearance cues in the recognition task. As indicated in Section 3.1, (semantic) context was heavily used by early computer vision systems as pre-defined rules in order to facilitate recognition. Propositional graphical models provide a simple way to model context since they can encode the structure of local dependencies in statistical frameworks. Employed at different semantic levels, they can include information about relative or absolute position of parts in an image. Graphical models have

Repr. \ Decision	Matching/Reasoning/Learning	Graphical Models	Other Statistical Approaches
Graphs	[Roberts, 1963] [Guzmán, 1971, Huffman, 1971] [Fischler and Elschlager, 1973] [Barrow and Popplestone, 1971] [Underwood and Coates, 1975] [Nevatia and Binford, 1977] [Shapiro and Haralick, 1982] [Grimson and Lozano-Pérez, 1987] [Keselman and Dickinson, 2005]	[Malisiewicz and Efros, 2009] [Galleguillos and Belongie, 2010] [Divvala et al., 2009] [Felzenszwalb et al., 2010] [Choi et al., 2012] [Nowozin et al., 2011] [Felzenszwalb and Huttenlocher, 2005] [Ramanan, 2011] [Damen and Hogg, 2009]	[Harchaoui and Bach, 2007] [Caetano et al., 2009] [Duchenne et al., 2011] [Lee and Grauman, 2012] [Li et al., 2012] [Yao and Fei-Fei, 2010] [Sridhar et al., 2010a] [Sridhar et al., 2010b]
Logic and Relational Languages	[Guzmán, 1968] [Tenenbaum and Barrow, 1977] [Shapiro, 1983] [Matsuyama and Hwang, 1985] [Dickinson and Davis, 1987] [Reiter and Mackworth, 1989] [Esposito et al., 1992, Malerba, 2003] [Russ et al., 1996] [Shanahan, 2005] [Hartz and Neumann, 2007] [Neumann and Möller, 2008] [Hartz et al., 2009] [D'Este and Sammut, 2008] [Farid and Sammut, 2012] [Dubba et al., 2010]	[Marton et al., 2009] [Xu and Petrou, 2009] [Tran and Davis, 2008]	[Yakimovsky and Feldman, 1973] [Berardi et al., 2004] [Gries et al., 2010]
Grammars	[Moayer and Fu, 1976] [González and Thomason, 1978] [You and Fu, 1979] [Pavlidis and Ali, 1979] [Fu, 1982] [Rosenfeld, 1982] [Koutsourakis et al., 2009]	[Zhu and Mumford, 2006] [Schmittwilken et al., 2009]	[Zhu and Yuille, 1996] [Lippow et al., 2008] [Han and Zhu, 2009] [Koutsourakis et al., 2009] [Lippow, 2010] [Girshick et al., 2011]
Compositional Hierarchies	[Hanson and Riseman, 1978] [Ohta et al., 1979] [Fu, 1983]	[Yang et al., 2008]	[Draper et al., 1989] [Fidler and Leonardis, 2007] [Yang et al., 2008] [Wang and Fei-Fei, 2006] [Epshtein and Ullman, 2007] [Sudderth et al., 2008] [Kreutzmann et al., 2009] [Fidler et al., 2009] [Hotz and Neumann, 2010]

Table 3.1: Axes for discussing old and new related work. Old papers are displayed in red while new ones in blue. The papers marked in black indicate the transition period. One can notice the dominance of red in the first column indicating that crisp logical and relational approaches were highly popular in early vision. Relational languages and logic-based approaches have been rarely used in modern computer vision, in crisp or probabilistic formulations, given the size of the computer vision community.

been successfully employed for visual recognition. A related survey is presented in [Galleguillos and Belongie, 2010].

Specific related work for visual recognition has exploited both directed and undirected graphical models. Directed graphical models express causal relationships between random variables. They model global distributions using local transition probabilities. Here fits the work of [Bar-Hillel and Weinshall, 2008], where a part-based object class model which uses appearance features, location and scale relations between parts is modeled via a star-like Bayesian network with nodes representing object parts and one hidden node which captures spatial and scale information. Parameter learning is done in a weakly supervised setting with a boosting approach where each part is treated like a weak classifier. Other recent approaches are presented in [Gupta and Davis, 2008, Karlinsky et al., 2010, Choi et al., 2012]. In [Gupta and Davis, 2008] a directed graphical model is used to estimate region labels and possible relationships between them in an Expectation-Maximization manner, while in [Choi et al., 2012] a tree graphical model is used to model object categories co-occurrences and spatial relationships between them. Both the tree structure and the parameters of the model are learned using maximum likelihood estimation. The work in [Karlinsky et al., 2010] is an extended directed star model.

Many object recognition approaches use undirected graphical models. Some of these are based on CRFs, a discriminative approach which estimate the conditional distribution and thus, have the ability to directly predict the labels. Some of the work on CRFs for visual recognition is presented in [Quattoni et al., 2004, Galleguillos and Belongie, 2010]. Other approaches employing CRFs to maximize object label agreement according to semantic and spatial relevance are proposed in [Galleguillos et al., 2008, Ladický et al., 2010, Nowozin et al., 2010, Nowozin et al., 2011].

Part-based models are popular undirected graphical models for visual recognition. They are currently among the state-of-the-art methods for general object recognition on different datasets. Their success is due to engineered feature representations, their structural representation and the discriminative algorithms employed. They are mainly extensions of the original pictorial structures introduced in [Fischler and Elschlager, 1973] and of their later statistical formulation proposed in [Felzenszwalb and Huttenlocher, 2005]. An overview of part-based models extensions and various learning methods to learn their parameters (such as maximum likelihood estimation, conditional random fields, structured max-margin models or latent-variable structural models) for human body recognition and its pose estimation is given in [Ramanan, 2011]. Additional extensions for object and scene recognition include [Fidler et al., 2013] and [Pandey and Lazebnik, 2011], respectively. Part-based models are rather powerful, although they can still handle limited variability in the structure.

Mixtures of part-based models can capture different object views or poses and can partially solve this problem [Felzenszwalb et al., 2010]. Constellation models are a special case of part-based models; they are fully connected graphical models [Fergus et al., 2007]. Although they are the most general, the assignment of features to parts becomes intractable for moderate numbers of parts. An undirected graphical model approach to region recognition using factor graphs trained in a generative manner is proposed in [Boutell et al., 2007].

Other graph-based statistical approaches In graph matching, patterns are modeled as graphs and pattern recognition techniques are used to find a correspondence between the nodes of different graphs. Graph matching techniques have made great progress from early vision. Statistically framed, recent work includes defining and learning kernels, distances or compatibility measures for graphs in order to solve visual recognition problems. In [Harchaoui and Bach, 2007, Lee and Grauman, 2012] graph kernels are used for image classification and object recognition, respectively, while in [Caetano et al., 2009, Duchenne et al., 2011] kernel, distance or similarity function learning is considered.

Several related approaches employed data mining techniques to detect higher-order groups of patches or objects that are related with each other spatially. Called also "grouplets" or "visual phrases", they are further employed as discriminative patterns to recognize activities [Yao and Fei-Fei, 2010], scenes and object categories [Li et al., 2012, Sadeghi and Farhadi, 2011, Pineda et al., 2009]. In addition to spatial relations, temporal ones are also used to recognize functional object categories from symbolic activity graphs in [Sridhar et al., 2008]. Constraint-based graph mining is employed to similar spatio-temporal activity graphs to discover in a supervised manner subgraph patterns that represent activity events [Sridhar et al., 2010a]. An unsupervised approach to relational graph mining of event patterns is proposed in [Sridhar et al., 2010b].

Relational and logical matching, learning and reasoning Graphical models can model context to some extent. They can handle limited variability in the structure due to an usually fixed number of parts and pair-wise relations, even when mixtures of models are employed. Additionally, they face computational costs due to their full grounding and lack of abstraction. More general in these respects are logic and higher arity symbolic graphs. Early work on crisp logic/graph matching and reasoning from 1970s and 1980s is described in Section 3.1, as Table 3.3.1 indicates. In fact, the early vision work was dominated by relational and logical approaches. This work was extended with compositional hierarchies at that time and, as described in the following paragraph, not so long ago it started to be framed in statistical (hierarchical) formulations.

However, original ideas on finding abstract graph models for recognition have been kept alive in few later works (e.g., [Russ et al., 1996]) and then revisited in modern visual recognition (e.g., [Keselman and Dickinson, 2005]). On the logic side, while formal logic has been used much in automated theorem proving and constraint satisfaction, its use in modern visual recognition has only recently started to rise again (see Table 3.3.1, first column, second row), but is limited in the large vision research community today. Nevertheless, there are still several (recent) inductive logic programming approaches extending early work with promising results in different application domains.

A promising logic-based approach from late 1990s was implemented by the system VEIL [Russ et al., 1996]. It was used to interpret aerial images of airports containing objects such as buildings, transportation networks, power transmission lines, trucks, natural terrain and so on. The framework was a layered architecture that separated the low-level extraction of visual primitives employing specialized data structures, from the declarative logic-based formal knowledge representation of symbolic structures used for reasoning. VEIL's goal was to enable the construction of explicit declarative visual models which included qualitative spatial, temporal and functional reasoning, flexible control of instance recognition and classification, and incremental scene processing. To this end the Loom language [MacGregor and Bates, 1987], a knowledge representation language based on description logic, was used. The tasks considered were object and event recognition. Its main achievement was to show that declarative representations and logic deduction as reasoning engine can be useful for image understanding. Some of these ideas were revisited 10 years later in the context of a realistic scenario for traffic intersection interpretation. In her work, [Hummel, 2010] employs description logic as representation language.

In document image analysis a relational distance between logical descriptions is learned from data [Esposito et al., 1992]. Using the same application domain as in [Malerba, 2003], an inductive logic programming system is employed to learn logical theories for document image understanding. The problem considered is that of mapping the layout structure of a document into its corresponding logical structure, that is its hierarchy of logical objects, such as sender/receiver of a letter, title, authors of an article, and so on. The mapping of the layout structure into the logical structure is, thus, represented as a set of rules.

Further, declarative and relational models of house facades concepts are estimated from logical interpretations of house images in [Hartz and Neumann, 2007]. As pointed out by the authors such descriptions of scene objects play an essential role in model-based scene interpretation. They show how ontological concept descriptions for spatially related objects and their aggregates can be learnt from positive and negative examples using version spaces and based on a relational description language that can capture both quantitative and

qualitative attributes and spatial relations. These concepts are integrated in a recognition control framework proposed in [Hartz et al., 2009], where learning of such concepts is one module. The use of description logics as a knowledge representation and reasoning language for high-level scene interpretation is demonstrated in [Neumann and Möller, 2008]. The aggregates composed of multiple parts and constrained by temporal and spatial relations are shown to be useful in representing object configurations, occurrences, events, and episodes that are required in different applications. Finally, the work in [D'Este and Sammut, 2008,Farid and Sammut, 2012] and [Dubba et al., 2010] use first-order clause inducing systems to learn classifiers for structured objects, concepts and activity events recognition. A first-order explanation-based reasoning approach to scene interpretation for robotics is employed in [Shanahan, 2005].

Nevertheless, these are crisp logic-based and declarative approaches that do not consider a probabilistic aspect of the recognition problem. One idea to improve visual recognition with relational approaches is to design better interfaces between the binary-valued logic and the probabilistic vision output. For example, this aspect is pointed out in [Esposito et al., 2001], where a rule inducer is employed to learn logical theories that are further used to recognize logical components of a document and for layout analysis. In the proposed framework the models are applied in a probabilistic formulation by extending the concept of Θ subsumption to probabilistic subsumption (or flexible-subsumption), which indicates the probability of A Θ-subsuming B.

Statistical relational learning SRL promises to improve visual recognition. Several first steps were made into this direction for hierarchical and non-hierachical setups in a static or dynamic setting [Cohn et al., 2008]. Although recent work mainly uses statistical formulations to solve recognition problems, as Table 3.3.1 indicates, in a relational context, they are mostly combined with hierarchical formalisms. In non-hierarchical setups, relational languages, graphs and logic-upgraded graphical models have been rarely employed by modern computer vision.

We next present some of the non-hierarchical approaches. An ahead of its time work was that by [Yakimovsky and Feldman, 1973], where the problems of image segmentation into meaningful regions and the recognition of these regions are considered. Bayesian decision theory is employed to integrate the high-level semantic and logic-based information into the reasoning mechanism for learning and inference procedures. Further, an integrated supervised approach for semantic structure extraction from biomedical multi-page document images can be found in [Berardi et al., 2004]. Images are first processed to extract both their layout and logical structures on the base of geometrical and spatial information. Then, textual content of logical components is employed for automatic semantic labeling of layout structures. To support the learning procedure a Naive Bayes

approach, among others, is employed.

The logical explanation-based approach of [Matsuyama and Hwang, 1985] for scene understanding is extended by a description logic-based approach for multimedia content in [Peraldi et al., 2009] and by a statistical formulation in [Gries et al., 2010]. The development, realized via Markov Logic Networks (MLNs), consists in a control strategy with respect to ranking image explanations according to their probabilities. Finding the best interpretations is controlled by generating an explanation only if the probability of the evidence being true is substantially increased. In [Marton et al., 2009] SRL techniques like MLNs and Bayesian Logic Networks (BLNs) are investigated to recognize kitchen objects in a robotics manipulation task by combining information from different sensor data. The models are trained in a supervised manner. Further, the hierarchical framework presented in [Petrou, 2008] exploits object functionalities and relations, expressed as logical rules, along with symbolic and semantical descriptions. MLNs are employed as SRL technique to express in softer ways such high-level descriptions [Xu and Petrou, 2009]. The work includes learning the structure of the network and the parameters of the model. Integrating MLNs into the hierarchical framework increases significantly the performance of labelling 3D representations of buildings, while the training examples are reduced significantly. MLNs have been also employed for visual event recognition in a parking lot [Tran and Davis, 2008].

Stochastic grammars Related hierarchical approaches framed in stochastic formulations are discussed hereafter. Crisp syntactic grammars employed in early vision were presented in Section 3.1. Notable progress in the use of grammars for image understanding has been made by stochastic attribute grammars [Zhu and Mumford, 2006, Zhu et al., 2007, Schmittwilken et al., 2009, Tylecek and Sara, 2011, Han and Zhu, 2009]. They model the hierarchical decomposition of scenes, objects, parts, primitives via terminal and non-terminal nodes and the context (spatial and functional relations) via horizontal links between the nodes. Therefore, the attribute grammar can be represented as an and-or graph, where the or-nodes give the possible alternatives in the hierarchy and the and-nodes are decomposed in components among which context constraints are enforced. It is a probabilistic context-sensitive grammar, where the probabilistic aspect is given by three components: the frequency of the branches in the or-nodes, the frequency of the primitives/ parts/objects and the frequency of their context constraints. Using the grammar one can build an and/or graph to interpret new images. The work in attributed grammars for scene understanding includes learning the parameters of the model [Porway et al., 2007a, Tylecek and Sara, 2011], the primitives [Tylecek and Sara, 2011], the relation set [Porway et al., 2007b] and the structure of the grammar [McAuley et al., 2009, Zhu et al., 2007]. Inference can be done in a bottom-up/top-down fashion using the grammars.

To solve the ambiguities in inferring labels of local primitives local constraints can be used [Han and Zhu, 2009, Schmittwilken et al., 2009].

Closely related to attribute grammars are probabilistic geometrical grammars (PGGs) [Lippow et al., 2008]. A difference from the attribute grammars is that they make stronger independence assumptions when modeling the context constraints, by assuming that the attributes of the parts are independent of its non-descendants, given the attributes of the parent, and of its descendants, given its children. These dependencies are modeled using Bayesian networks and require fewer parameters, which increases the speed of learning and improves performance. PGGs do not allow primitives as internal nodes. The independence assumptions which impose a partial order generates a parse tree as image interpretation for an image, while in an attribute grammar this is represented by a parse graph due to the horizontal links.

Other related work in the context of grammars is the work by [Siskind et al., 2007], where a stochastic-context-sensitive grammar is learnt generatively in a supervised way to distinguish between images of cars and images of houses. This differs from [Zhu and Mumford, 2006] and [Lippow et al., 2008] in that it models the segmentation of the entire image and not only few object classes across the hierarchy. In [Girshick et al., 2011] an object detection grammar is defined. It is a nondeterministic grammar assigning weights to the grammar rules. The early deterministic shape grammars are revisited by recent work to automatically derive 3D models of highly symmetric urban buildings from single facade images. In [Koutsourakis et al., 2009] the grammar contains rules which describe how basic shapes like floor, tile and small regions in the tiles interact to produce more complex geometries. The parameters of the grammar are floor, tile and minimum region sizes and are estimated from training data. [Müller et al., 2007] optimizes the shape grammar parameters (e.g., window height) in an unsupervised manner. It does not, however, express complex spatial and context constraints.

Probabilistic compositional hierarchies Grammar models have considerable representational power. However, they must always find a tradeoff between the representational and computational performance. For example, the work in [Lippow et al., 2008] exploits statistical independence assumptions between different parts in order to do so. Therefore, in the search for computational efficiency in structured probabilistic representations, some visual recognition researchers have focused on the principle of hierarchical compositionality to build structured and stochastic models of images and objects. This idea was pursued starting from early vision. The work on compositional hierarchies in that period culminated with a stochastic extension of the VISIONS system, which was refined for more than a decade [Draper et al., 1989]. It was continued by modern machine vision research via several frameworks.

In [Zhu et al., 2011], such models are denoted recursive compositional models (RCMs). They represent visual patterns in a hierarchical way, such that more complex structures are composed of more elementary ones. In addition, they are usually probabilistic and the probabilities are defined over these structures in the hierarchy. This compositional structure allows to exploit both the structure of the problem and computational efficiency. Keeping the hierarchical and generative aspects of grammars, they are different from grammars, by considering overlapping situations of reusable parts and treating dependencies among different parts of a parent node, also called aggregate/composition, in a non-Markovian fashion. They are graph-structured networks which allow arbitrary dependencies among the children and impose distributions on hierarchical representations of the image. As a result, long range spatial relations between elements of the scene can be represented by local short-range dependencies, given the hierarchical decomposition. Also, their recursive compositional structure means that when modeling multiple objects, parts between different object models can be shared. Thus, the compositional nature of RCMs enables efficient inference and learning algorithms. Inference is performed by searching for probable states of the sub-parts and using them to propose states for the parts. Similarly, learning exploits the recursive structure by first learning sub-parts and then learning ways to combine them to form larger parts. Their use includes object detection, object parsing, object matching, and image tagging. Their effectiveness was shown on several benchmarks [Fidler et al., 2009, Li et al., 2009, Zhu et al., 2008, Zhu et al., 2012]. Learning the graph structure of a RCM in an unsupervised manner is presented in [Fidler and Leonardis, 2007] for a deterministic setting. In [Epshtein and Ullman, 2007] a hierarchy of visual features is built to recognize semantic image parts.

Further, in [Wang and Fei-Fei, 2006] a latent theme variable layer is introduced and shared among categories and images to build a hierarchy of semantic concepts. It captures the inter-dependency between the patches by using semantic linkage (e.g., if they occur together or not). A semi-supervised learning approach of hierarchical Dirichlet process is employed to learn the semantic taxonomy of the model. Another hierarchical Dirichlet process extension is presented in [Sudderth et al., 2008]. In [Ommer and Buhmann, 2010] the compositions are shape models whose configurations are based on features like edges, pixel intensities, relative orientation and location. The paper proposes an unsupervised learning approach to parameter estimation for the hierarchical compositional model. In [Parikh and Chen, 2007] a tree-shaped hierarchy is presented. The tree groups semantically related objects based on similar part-part/object-object spatial distance and appearance features for particular scene categories (e.g., office). The structure of the hierarchy is learnt in an unsupervised way by clustering extracted features. The approach can afterwards detect parts of the image that belong to the foreground objects,

cluster these parts to represent objects and provide an understanding of the scene by hierarchically clustering these objects.

There are other compositional hierarchy frameworks that exploit spatial relationships between objects, however not many are combined with logic. An exception is the tower of knowledge (ToK) architecture [Petrou, 2008]. It has 4 levels: the image level (where all the feature measurements are available e.g. spatial relations, colour), the semantic level (the object class labels) and on top, encoded in a logical form, are the functional and description layers. The last two layers give feedback in a top-down manner, by checking measurement descriptions and imposing constraints on these descriptions. Another exception is the work by [Kreutzmann et al., 2009], where the context exploited during object recognition is created step-by-step from evolving image interpretations based on the objects recognized at some point in time. Therefore, the inference process allows an evolving context as prior. The logical language used in the framework is description logic. A rule-based representation is proposed in [Bohlken and Neumann, 2009] as an alternative to solve the limitations of description logic. Finally, the framework proposed in [Hotz and Neumann, 2010] relies on a Bayesian tree-shaped hierarchy where each aggregate node is expressed in logical form. Except the "part of" relationship and qualitative spatial relations such as "below neighbour of", it allows for quantitative properties, such as shape, height and width, number of part aggregates and their class. A supervised learning approach of the hierarchy is presented in [Hartz, 2009], where probabilistic structure graphs for each level of the hierarchy are obtained by finding the maximum common attributed subgraph from the graphs given in the training phase.

In [Yang et al., 2008] a bottom-up and top-down hierarchical structure is used to classify an object at different levels of the hierarchy. To this end, a different part-based star model that capture spatial information between parts is employed at each layer of the hierarchy. Another layered approach to object categorization is proposed in [Bouchard, 2005].

3.4 Conclusions

We have presented a brief history of early and modern vision from a relational representation point of view. We showed that relational representations were popular in vision recognition 30-40 years ago. The research focus at the time was to design the high-level relational background knowledge, the representation of the image data, the reasoning or control mechanisms, and the use of the data and knowledge in the control scheme. We also motivated why the interest

was later shifted towards statistical pattern recognition. Today, relational representations are rarely used in computer vision. In one talk Kanade notes:

> *"...Successes in computer vision primarily solve problems constructing geometrical world-models from image data, whereas the important problem of relating visual images to high-level scene descriptions is largely neglected ..." Takeo Kanade , 2003*

However, now, different from then, much progress has been made concerning the drawbacks of those times. Low-and mid-level vision procedures are much more mature. Statistical machine learning has made an immense progress and a plethora of robust and very efficient techniques are available. These developments are potentially enough to re-investigate old ideas on relational vision and to support ambitious relational/logical representations and goals. One example is the recent work on hierarchical models by Zhu and Mumford, which re-visits the early work by Barrow and Tenenbaum. Now, the old ideas on filtering are implemented using more modern statistical techniques such as particle filtering and Markov Random Fields. Another example is the work by [Girshick et al., 2011], which employs soft attributed grammars to obtain a state-of-the-art object detector in 2011.

Therefore, given the advancement of data availability, computational power, the maturity of low-and mid-level features and statistical methods, efforts need to be invested now in advancing representational and learning techniques for visual recognition. Chapter 2 shows the benefits of SRL techniques, which combine the advantages of relational representations with those of statistical learning. Recent successes in employing SRL techniques to other domains, such as bioinformatics [Kimmig and Costa, 2012] and natural language processing [Verbeke et al., 2012], motivates us to reinvestigate their use for visual recognition tasks. As one could conclude from Section 3.3, in computer vision they have been only timidly used so far. Thus, we believe it is time to transform SRL for visual recognition into an active research area. This idea is encouraged by several quite successful attempts already made, which are described in Section 3.3. The main difficulty in directly employing current SRL techniques for visual recognition is the large representational and semantic gap between the SRL formal frameworks and the low-level primitives. Practically, this gap depends from domain to domain and from problem to problem. Nevertheless, there are still few approaches that have tried SRL for visual recognition and much potential for further investigation is left. In this thesis we contribute a few [Antanas et al., 2009, Antanas et al., 2012a, Antanas et al., 2013a, Antanas et al., 2014].

If SRL succeeds to accomplish the dream of early vision, we can further hope to have general vision systems that are domain independent and can incorporate any

kind of world knowledge, and thus, obtain truly intelligent machine vision. The generalization ability of SRL and its advantages seem to make SRL a theoretically sound approach to solve some of the control problems. However, another important aspect is designing integrated frameworks which can customize the union of different techniques for different problems. This issue raises from the fact that SRL techniques may work for a certain category of structured problems, while for other problems, that are more appearance-based, purely vector-form representations and statistical techniques may be more suitable. This implies finding the best sequence of recognition steps, or recognition strategy, to obtain the final goal. How to structure such an integration is still a major open problem. Logic can, however, provide a representation suitable for specifying and guiding the processing of visual information, in which bottom up construction of complex objects from their constituents can be flexibly combined with top down reasoning for the constituents of an object whose existence is hypothesised. In the research process, it is important to always keep in mind and revisit the lessons already learned.

Chapter 4

Understanding Images of Houses Relationally

As indicated in the previous chapters, many of the approaches tackling image understanding problems heavily rely on dense appearance cues in the form of low and mid-level features [Tuytelaars and Mikolajczyk, 2007, Felzenszwalb et al., 2010, Fergus et al., 2007]. However, looking at Figure 4.1, house facades especially exhibit considerable structure that can be captured using qualitative spatial relations. In this case, it is more intuitive to understand and describe a visual scene in terms of hierarchical *structural* or *graph-like* representations, which express the natural composition of scenes into *objects*, *parts of objects* and lower-level *substructures* [Pinz et al., 2009]. For example, a typical house consists of aligned elements such as: a roof, some windows, one or more doors and possibly a chimney. A hierarchical aspect is that a window and a chimney themselves are composed of particular configurations of local features (e.g., corners with a certain appearance arranged in a rectangular- like way and 'brick'-like patterns of a certain shape, respectively).

This chapter puts relational representations to work. To this end, it addresses the problem of *hierarchical understanding of images of houses*. For each layer in the hierarchy we make use of qualitative spatial relations to detect and classify substructures in images. Next, we employ them one layer up the hierarchy to obtain even higher-level semantic structures. For our application of street view images we utilize a four-layer hierarchy in which subsequently corners, windows and doors, and individual houses are detected. We present two statistical and relational approaches to the image understanding problem: a distance-based technique and a kernel-based one.

Figure 4.1: Examples of house facades in Eindhoven. The third image from left to right is a house facade annotated with windows, doors and individual houses.

Instead of using a formal model of the distribution of scenes (e.g., in the form of a grammar), the *relational distance-based* approach employs recent developments from relational learning. Yet, our contribution preserves desired properties of grammars, that is, it employs structured input features and outputs a structured explanation of the image at each layer in the hierarchy. We show how theoretical results in relational distance metrics [De Raedt and Ramon, 2009] can be utilized as a generalization technique to help recognize higher-level structures in an image. The result is a framework in which spatial configurations and relational distance functions are used throughout all levels of a hierarchy, in a unified way, to recognize objects.

Motivated by our results on using distances between logical interpretations of images to hierarchically detect known structures, our second approach replaces the relational distance functions with *kernels for graphs*. The resulting approach is more principled and more robust to noise than the relational distance, as it is grounded in a statistical learning framework. Furthermore, it is computationally more tractable and provides improved results. Our earlier approach relied on more expensive logical matching and generalization operations and was more tailored towards the house facade application.

The two statistical relational learning approaches build on ideas from statistics to address uncertainty, while incorporating a relational representation of the problem. Images are described in terms of automatically extracted semantic parts and relationships between them, thus as relational databases or (hyper)-graphs. Domain knowledge can easily be incorporated using logical rules. Furthermore, the declarative approach offers a flexible and interpretable way to consider both appearance and spatial information in an image.

The chapter is structured as follows. Section 4.1 gives an overview of the hierarchical framework. Next, we present the visual primitives and how they were obtained at the base layer in Section 4.2. Section 4.3 describes the representation at each layer and the learning problem. We explain the distance and kernel approaches in Sections 4.4 and 4.5, respectively. The empirical

evaluation is presented in Section 4.7. Finally, Section 4.8 presents related work and we conclude in Section 4.9.

An early version of most work in this chapter was published in

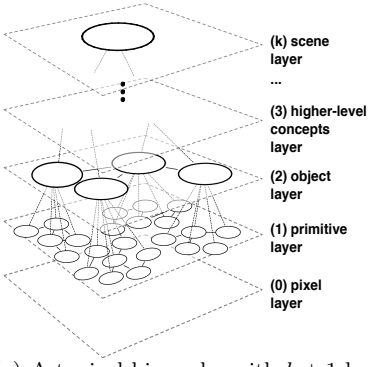
- Antanas, L., van Otterlo, M., Oramas Mogrovejo, J. A., Tuytelaars, T., and De Raedt, L. "A relational distance-based framework for hierarchical image understanding". In: *Proceedings of the 1st International Conference on Pattern Recognition - Applications and Methods*, 2012.
- Antanas, L., Frasconi, P., Costa, F., Tuytelaars, T., and De Raedt, L. "A relational kernel-based framework for hierarchical image understanding". In: *Structural, Syntactic, and Statistical Pattern Recognition*, Lecture Notes in Computer Science, Springer, 2012.
- Antanas, L., van Otterlo, M., Oramas M., J. A., Tuytelaars, T., and De Raedt, L. "There are plenty of places like home: using relational representations in hierarchies for distance-based image understanding". In: *Neurocomputing Journal*, 2013.

4.1 The Hierarchical Framework

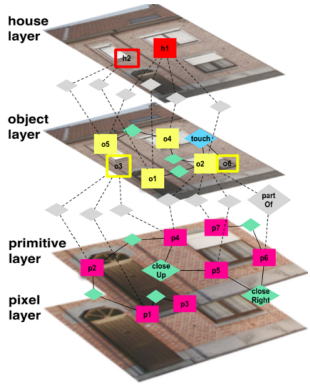
In our hierarchical framework, an image Z is described at several layers $0, \ldots, k$ in the hierarchy, with 0 the base layer and k the top layer. At each layer, the description consists of a set of classified regions of interest or parts C_i, their properties as well as the attributed spatial relationships among them. In Figure 4.2(a) parts are indicated by a circle, while the relations between them with an edge. The classes denote the concepts the parts belong to. The task then consists of using the description of an image at layer i to generate and classify regions of interest C_{i+1} at the next higher level $i+1$ in the hierarchy.

We assume that manually labeled examples of object categories we want to recognize throughout all layers in the hierarchy (i.e., houses, windows and doors) are available as training data (Figure 4.1). Each house in the training set is annotated with the locations and shapes of its constituent windows and doors. We represent an object as a set of parts and a set of qualitative spatial relations defined on them (hence; a relational attribute graph). Each image substructure is spatially embedded in a 2D plane, and parts are related to each other with respect to this space.

Figure 4.2(b) shows the hierarchical structure of a partial house facade with all four layers. In this hierarchy, the *pixel layer* consists of the image itself.



(a) A typical hierarchy with $k+1$ layers. A layer i is a set of classified entities (empty circles) arranged in spatial configurations. Each configuration generates a classified entity at the next higher-level $i+1$ in the hierarchy.



(b) The hierarchy mapped to the house facades application. The result is a 4-layer hierarchy: pixel, primitive, object and house layers.

Figure 4.2: The hierarchical framework.

In the *primitive layer* the parts are pixels and the regions of interest to be recognized are *local patterns*, i.e., corners and edges. The *object layer* is then built from spatial configurations of such local patterns with their properties, forming regions of interest belonging to concepts such as *door* and *window*. These then become parts at the next level and are used to find higher-level regions of interest representing *houses*. We stop at the *scene layer* which groups houses into streets. As an example, the object layer consists of corner primitives detected at the primitive layer with their properties, spatial configurations of corners and candidate windows and doors. The task is to determine the classes the candidates belong to.

This hierarchical image understanding framework propagates the detected parts in a *bottom-up* manner through each layer. Information flow is similar at all levels. The layer-wise algorithm is based on three main steps. First, recognized parts C_{i-1} from the previous layer are used to generate a relational representation of the problem at the current layer. Current-level candidate parts are included in the layer representation and are generated using configurations of C_{i-1}. Next, candidates are evaluated by the classifier. Finally, only the best C_i's are selected to be further employed at the next layer.

Figure 4.3 illustrates the information flow at one layer in the framework. The classification step consists of the relational kernel/distance and the statistical/instance-based learning. While the distance works directly on the relational features via logical matching, the kernel-based approach works on

a graphicalized representation of the relational representation. The selection step uses extra contextual constraints to keep detected regions of interest that best explain the image. Although, in theory, the selection step can be used in both approaches, in practice, we use it only for the distance-based approach, as the kernel can naturally capture contextual information. After we obtain the set of selected candidates, a final post processing step is applied. In the following sections we explain how an image is relationally represented layer-wise and how our recognition problem is formalized and modeled within a relational representation.

4.2 From Images to Visual Primitives

The goal of this section is to describe the visual primitives which are used to build the relational representation at each layer. The primitive layer is an exception, as instead of our SRL approaches, we employ standard computer vision routines. It takes as input image pixels and groups them in corner-like features. Each corner is characterized by a pair of edges with their coordinates. Their intersection generates the centre of the corner. Corners, together with their properties represent image parts at the object layer together with their attributes. We employ the 2AS interest point detector [Ferrari et al., 2008] to detect corners formed by chains of 2 connected, roughly straight contour segments. Each corner-like part can be one of the types in the set $\{topR, topL, botR, botL\}$ representing top-right, top-left, bottom-right and bottom-left corners, respectively. The corner type is given by the orientation of the segments composing the 2AS. Because we can get many detections we only keep square-like corners with an angle of $\approx 90°$. Figure 4.4(a) illustrates selected corner-like detections on a test image.

To avoid redundant corners and thus, increased computational cost, we collapse almost overlapping corners of the same type. Further, to discard irrelevant corners found on other structures than buildings (e.g., trees), we train a binary classifier using features describing the corners. For this purpose, we use the HOG descriptor [Dalal and Triggs, 2005] to characterize the appearance information of each corner. In practice we use a variation of the HOG descriptor with 16 orientation bins, window size of 128x128 pixels and a block size of 8x8 cells that showed improved results. We use the training annotations of windows and doors to label the detected corner instances characterized by the feature descriptor.

At the object layer, visual primitives are the sparsely detected corners and their associated edges. Instead of a raw feature descriptor characterizing each corner primitive, we train another classifier to map each HOG to a discrete attribute,

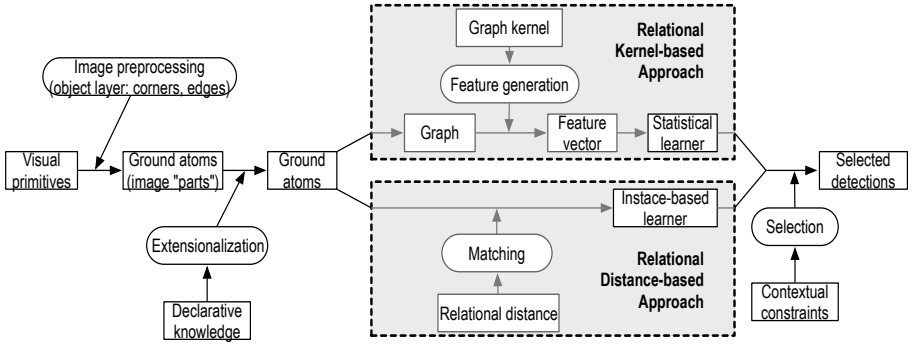


Figure 4.3: Information flow at one layer: detection of visual primitives, relational representation and declarative feature construction, relational distance/kernel module, statistical learner and regions selection.

either a *window* or a *door* label. From this point of view, we partly assume a *discrete* representation of the recognition problem at the object layer. Additional properties of a corner primitive are the corner type, its estimated bounding box and edges. Figure 4.4(b) illustrates classified corner-like detections together with their properties.

At the house layer, visual primitives are doors and windows found at the object layer and possible *houses* are candidate configurations of objects. Attributes of visual primitives at this layer are the labels *door* and *window* and their estimated bounding box.

4.3 Relational Problem Formulation

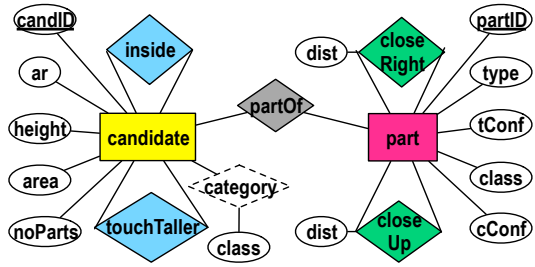
Let us now describe how we represent relationally an image Z at the object and house layers in the hierarchy. We assume knowledge about the identity of the layer and access to automatically detected and extracted parts in the image at this layer, together with their properties. Based on these assumptions we define and employ a *relational language* which is derived from its associated E/R data model and thus, is based on entities, relationships linking entities and attributes that describe objects and their relationships. Figure 4.5 shows the E/R diagrams for our problem at both layers. They provide an abstract representation of the examples for the class of interest (i.e., *window* and *door* at the object layer, and *house* at the house layer).

(a) Pre-selected corner-like detections on a test image. Estimated corners are marked by circle-cross pairs. The corner type is indicated by the color combination of the pair. Plain crosses mark the edges endings.

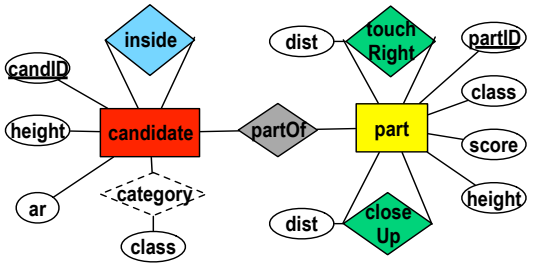
(b) Classified corner-like detections on the same test image. Red indicates corners classified as belonging to a window, while green those classified as belonging to a door.

Figure 4.4: Examples of corner detections in an image at the primitive layer.

The language is house facade domain specific and allows to specify, in a declarative way, relational features. It consists of visual entities, candidate entities, spatial relationships between entities, and member relationships, all



(a) E/R model at the object layer.



(b) E/R model at the house layer.

Figure 4.5: E/R diagram.

with their attributes. The language slightly differs from one layer to another, depending on the characteristics of the entities at each layer. Entities and relationships correspond to tuples (or relations) in the database. They can be visualized as relational facts, in Figure 4.6 for the object layer and in Figure 4.7 for the house layer.

A *visual entity* represents a part or region of interest of the image at the current layer i. It is represented by the relation $\texttt{part}(id, attr_1, \ldots, attr_n)$, which indicates that each visual entity has a unique identifier id (underlined oval) and several attributes $attr_i$. At the object layer, the tuple $\texttt{part}(p_1, botL, 0.62, door, 0.79)$ specifies a part entity (depicted purple in Figure 4.5(a)), where p_1 is its identifier and the other arguments are properties extracted by the previous layer in the hierarchy. They are the corner type, its detection confidence, the corner category and its associated class confidence. At the house layer an example of a visual entity (depicted yellow in Figure 4.5(b)) is $\texttt{part}(o_1, door, 1.5, tall)$, where o_1 is the part identifier and the rest are its properties, that is the object category (door), its detection score from the object layer and its discretized height.

A *candidate entity* represents a possible concept of interest to be recognized

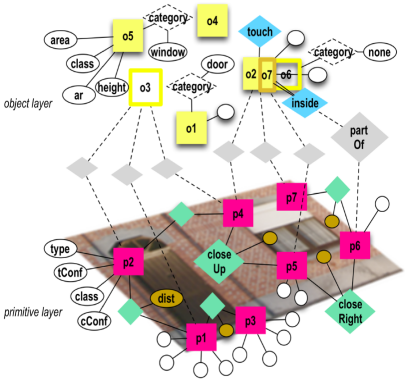
and is a candidate visual entity at the next level $i + 1$. It is represented by the relation $\texttt{candidate}(ic, attr_1, \ldots, attr_n)$, indicating that each visual entity has a unique identifier `ic` (underlined oval) and several attributes $attr_i$. Semantically, it groups a set of visual entities and their spatial configuration into an example or learning instance. At the object layer, the tuple $\texttt{candidate}(o_1, thin, tall, area3, 4)$ represents a possible object of interest (yellow in Figure 4.5(a)). It has identifier o_1 and properties describing its discretized aspect ratio, height, area and the number of visual entities it groups. At the house layer a candidate entity (red in Figure 4.5(b)) represents a house candidate and is visualized relationally as a tuple $\texttt{candidate}(h_1, ar6, tall4)$, where, except for the candidate identifier h_1, the other two arguments are the discretized aspect ratio and height. Size-related properties are estimated from the extracted bounding box of the candidate based on the visual entities it groups.

Spatial relationships impose a structure on entities (e.g., spatial neighborhood) and are linked to the entities that participate in the relationships. In our problem, we have spatial relationships amongst visual entities (green diamonds) and, respectively, amongst candidate entities (blue diamonds). They are derived from the spatial localization of the entities, i.e., bounding boxes, and extension. An example is the relationship $\texttt{closeRight}(p_3, p_1, 232.6)$, which indicates that visual entities p_3 and p_1 are spatially close to each other and aligned on the X axis with p_3 to the right of p_1. It has as attribute the Euclidian distance between the bounding boxes centres. Other spatial relationships are defined in Section 4.3.1.

Additional relationships are introduced by the predicate $\texttt{category}(ic, class)$ (white diamond), which is linked to candidate entities and indicates the category of the candidate, and the membership relationship $\texttt{partof}(id, ic)$ (grey diamonds), which links visual entities to candidate entities.

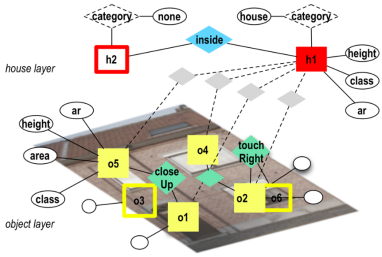
In practice, the language is specified using the kLog engine for the kernel-based approach [Frasconi et al., 2012] and using a self-developed framework for the distance-based approach. Nevertheless, both implementations closely follow the relational language introduced above and are embedded in Prolog. They only differ with respect to some implementation assumptions. While kLog is a general purpose relational language for kernel-based learning that allows to specify different learning problems, the self-developed framework is tailored towards the houses application using specific defined distance functions. In addition, kLog has a more principled and general syntax than the distance-based framework.

A clear advantage of the relational language is that it supports both extensional as well as intensional relations. Extensional relations are explicitly listed sets



$x = \{\texttt{part}(p_1, botL, door), \texttt{part}(p_2, topL, door),$
$\ldots, \texttt{part}(p_5, botL, win), \texttt{part}(p_6, botR, win),$
$\texttt{part}(p_7, topR, win), \texttt{closeUp}(p_2, p_1, 226),$
$\texttt{closeRight}(p_3, p_1, 145), \texttt{closeRight}(p_6, p_5, 161),$
$\texttt{closeRight}(p_4, p_2, 312), \ldots,$
$\texttt{candidate}(o_1, thin, h5, area3, 3),$
$\texttt{candidate}(o_2, squared, h3, area3, 3),$
$\texttt{candidate}(o_3, squared, h5, area5, 3),$
$\texttt{candidate}(o_5, thin, h3, area2, 5), \ldots, \texttt{partOf}(p_1, o_1),$
$\texttt{partOf}(p_2, o_1), \texttt{partOf}(p_3, o_1), \texttt{partOf}(p_4, o_2),$
$\texttt{partOf}(p_5, o_2), \texttt{partOf}(p_6, o_2), \texttt{partOf}(p_7, o_2), \ldots,$
$\texttt{inside}(o_7, o_2), \texttt{touch}(o_6, o_2), \ldots\}.$
$y = \{\texttt{category}(o_1, door), \texttt{category}(o_5, window),$
$\texttt{category}(o_3, none), \ldots\}.$

Figure 4.6: Description of the house facade image at the object layer. Entities are purple/yellow squares, relationships are diamonds (green/blue for spatial/functional constraints, grey for membership constraints), properties are circles. Candidate entities not belonging to a class of interest are empty squares. A visual interpretation $i = (x, y)$ is on the right; x specifies the input features, while y is the learning target.



$x = \{\texttt{part}(o_1, door, 1.56, h5), \texttt{part}(o_2, window, 1.7, h4),$
$\texttt{part}(o_3, window, 0.5, h2), \texttt{part}(o_5, window, 2.1, h3),$
$\texttt{part}(o_6, window, 0.8, h4), \ldots, \texttt{closeUp}(o_5, o_1, 450),$
$\texttt{closeUp}(o_4, o_2, 390), \texttt{touchRight}(o_6, o_2, 134), \ldots,$
$\texttt{candidate}(h_1, ar6, tall4), \texttt{candidate}(h_2, ar2, tall4),$
$\ldots, \texttt{partOf}(o_5, h_1), \texttt{partOf}(o_1, h_1), \texttt{partOf}(o_4, h_1),$
$\texttt{partOf}(o_2, h_1), \ldots, \texttt{inside}(h_2, h_1)\}.$
$y = \{\texttt{category}(h_1, house), \texttt{category}(h_2, none), \ldots\}.$

Figure 4.7: A description of a facade image at the house layer. Entities are yellow/red squares, the rest is kept the same as for the object layer.

of given facts, whereas intensional relations are defined implicitly using logical rules in Prolog. In other words, intensional relations can be derived from other extensional or intensional relations given a set of rules. They represent domain-related feature construction and those used in our problem are described in the next section. In our case extensional relations are those introduced by visual entities, while the rest are intensional.

4.3.1 Declarative and Relational Feature Construction

Intensional facts are spatial relations and candidate entities. The grounding of intensional relations is computed using Prolog's deduction mechanism and represents the extensionalization step at one layer in Figure 4.3.

Spatial Relations

The spatial relations used are `closeUp`/3, `closeRight`/3, `touchRight`/3, `inside`/2 and `touchTaller`/2. They link visual entities or candidate entities and are derived using notions of spatial theory from spatial localization (or bounding boxes) of the entities. Spatial relations are defined using a logical background knowledge or a set of Prolog rules. Example 13 shows how the spatial relation `closeRight`/3 is defined as a logical rule. In practice, the rule is implemented as Horn clauses in Prolog.

***Example* 13.** *The relation* `closeRight`/3 *is defined as follows:*

$$\texttt{closeRight}(E_1, E_2, Dist) \leftarrow \texttt{part}(E_1, \_, \_, \_, \_),\ \texttt{part}(E_2, \_, \_, \_, \_),\ \texttt{rightOf}(E_1, E_2),\ \texttt{closeBy}(E_1, E_2, Dist).$$

The relation is not symmetric and directionality is imposed by the `rightOf`/2 *condition. Predicates* `closeBy`/3 *and* `rightOf`/2 *are defined in the following way:*

$$\texttt{closeBy}(E_1, E2, Dist) \leftarrow \texttt{bb}(E_1, BB_1),\ \texttt{bb}(E_2, BB_2),\ \texttt{edist}(BB_1, BB_2, Dist),\ Dist < \epsilon.$$

$$\texttt{rightOf}(E_1, E_2) \leftarrow \texttt{right\_fuzzy}(E_1, E_2),\ not(\texttt{up\_fuzzy}(E_1, E_2)),\ not(\texttt{down\_fuzzy}(E_1, E_2)),\ not(E_1 == E_2).$$

based on the visual entities bounding boxes BB_i. *Finally, the predicate* `right_fuzzy`/2 *is specified as:*

$$\texttt{right\_fuzzy}(E_1, E_2) \leftarrow \texttt{bb}(E_1, BB_1), \texttt{bb}(E_2, BB_2), \texttt{getMinX}(BB_2, X^2_{min}), \texttt{getMaxX}(BB_2, X^2_{max}), \texttt{getMinX}(BB_1, X^1_{min}), Width\ is\ X^2_{max} - X^2_{min}, X_{fuzzy}\ is\ X^1_{min} - z \cdot Width, X^1_{min} \geq X_{fuzzy}.$$

In words, E_1 is close to the right of E_2 if both spatial constraints are fulfilled. E_1 is right of E_2 if the min and max X coordinates of BB_1 are smaller than the minimum and the maximum X coordinates of BB_2 in a fuzzy way (as defined by the predicate `right_fuzzy`), and if E_1 is not too much above or below of E_2, also in a fuzzy way (as defined by similar predicates `up_fuzzy` and

`down_fuzzy`). Predicate `bb/2` returns the bounding box BB_i of an entity with id E_i. The fuzzy definition of the predicate `right_fuzzy` is illustrated in the example above. In words, it checks if the most left X coordinate of the bounding box of E_1 is greater or equal than a fuzzy variant of the most right X coordinate of the bounding box of E_1, such that a spatial overlap with E_2 on its right side is possible. Thus, the fuzzy definition allows a spatial displacement relative to the width ($Width$) of E_2 with a certain degree measured by the constant factor z. The predicate `getMinX/2` returns the X coordinate of the most left corner of the bounding box that it takes as input parameter. Finally, `edist/3` unifies the variable $Dist$ with the Euclidian distance between the two bounding boxes, and is thresholded to obtain the `closeBy/2` relation. The threshold ϵ is defined relatively to the size of the objects (at the house layer) or is fixed (at the object layer) and estimated experimentally from the training data. While the spatial relation `closeUp/3` is defined in a similar way as `closeRight/3`, but on the Y axis, `touchRight/2` indicates if two entities are spatially touching.

Another example is the relation `inside/2`. It holds if one entity is spatially inside another. This relationship is used both at the object layer (amongst likely window candidates) and at the house layer (amongst houses candidates). Thus, there are small variations in the definitions for the two layers. Example 14 illustrates one definition.

***Example* 14.** *The relation* `inside/2` *at the house layer is defined as follows:*

$$\begin{aligned}\texttt{inside}(C_1, C_2) \leftarrow\ & \texttt{bb}(C_1, [X^1_{min}, X^1_{max}, Y^1_{min}, Y^1_{max}]),\\ & \texttt{bb}(C_2, [X^2_{min}, X^2_{max}, Y^2_{min}, Y^2_{max}]),\\ & X^1_{min} > X^2_{min}, X^1_{max} < X^2_{max}, Y^1_{min} > Y^2_{min}, Y^1_{max} < Y^2_{max}.\end{aligned}$$

where the inequalities are evaluated on similar fuzzy variants of the bounding boxes.

A final example (illustrated in Example 15) is the spatial-functional relation `touchRightTaller/2`, which is defined at the object layer between candidate entities. In words, the relationships holds if a candidate entity C_1 is touching to the right another candidate entity C_2 and is considerably taller.

***Example* 15.** *The relation* `touchRTaller/2` *is defined as follows:*

$$\begin{aligned}\texttt{touchRTaller}(C_1, C_2) \leftarrow\ & \texttt{candidate}(C_1, \_, \_, \_, \_),\ \texttt{candidate}(C_2, \_, \_, \_, \_),\\ & \texttt{touchRight}(C_1, C_2),\ \texttt{height}(C_1, H_1),\ \texttt{height}(C_2, H_2),\\ & H_1 > f \cdot H_2,\ \texttt{hasDoorPart}(C_1),\ \texttt{hasDoorPart}(C_2).\end{aligned}$$

where the constraint `hasDoorPart/1` *ensures that the candidate entity considered in the relationships groups at least one visual entity belonging to a door. This improves door recognition, as it enforces a constraint on the height of the*

candidates belonging to the same facade that may be doors. The constraint is reflected via the height factor f.[1]

Candidate Entities

Similar to spatial relations, candidate entities are estimated in a declarative way as intensional relations from the bounding boxes of visual entities grouped by the candidate entity. The generation of meaningful new entities is also a novel task in the relational learning context. It can be seen as a dual to *predicate invention* [Muggleton and Buntine, 1988]. There the goal is to determine new and useful predicates. Here the task is to invent new entities. In a probabilistic context, it is related to *existence uncertainty*, a term coined in the literature on probabilistic relational models [Getoor et al., 2000]. In practice, we consider different definitions of candidate entities, depending on the layer in the hierarchy.

***Example* 16.** *The candidate relation candidate/5 at the object layer generated by 3 visual entities is defined as follows:*

$$\begin{aligned}\texttt{candidate}(Id, Ar, A, H, NoP) \leftarrow\ & \texttt{sprl}(A, B), \texttt{sprl}(B, C), \texttt{edge}(E_{ab}, A),\\ & \texttt{edge}(E_{ab}, B,), \texttt{edge}(E_{bc}, B), \texttt{edge}(E_{bc}, C),\\ & \texttt{getProp}([A, B, C], Ar, A, H, NoP, EntList),\\ & \texttt{getId}(EntList, Id).\end{aligned}$$

Here `sprl`*/2 brings together the pairs of parts that satisfy any of the spatial relations* {`closeRight`*/3,* `closeUp`*/3*} *and that share an edge*[2]*;* `getProp`*/6 calculates the discretized properties of the candidate relation, i.e., aspect ratio, area, height, total number and list of visual entities characterizing the bounding box of the candidate;* `getId`*/2 associates a unique identifier to the newly generated candidate, based on the combination of visual entities identifiers.*

At the object layer we consider also candidates generated by 4 parts that satisfy the square-like spatial constraint. However, these parts, together with their bounding boxes, determine the spatial frame (or bounding box) of the candidate. The definition of `getProp`*/6* considers inner entities as well as those along the sides of visual entities with respect to this frame. At the house layer, the edge condition from the definition is removed and the `sprl`*/2* constraint belongs to the extended set {`closeRight`*/3,* `closeUp`*/3,* `touchRight`*/3*}. To find the houses in case of noisy information (e.g., when only part of the house is visible)

[1] One can define a similar relation `touchLTaller` which holds if an entity is touching to the left another entity and is considerably taller.

[2] In practice it is a softer condition that checks if the edge passes through one of two corners' bounding box in a fuzzy way and in a strict way through the other.

and thus, the best explanation of the image in the selection step, the number of visual entities used for candidate generation varies from 2 up to 6. The limits on the number of visual entities are estimated from the training data. For example, if the image contains some parts of a (hypothetical) house, they can be regarded as configurations on their own (e.g. the partial house at the end of the facade on the left in Figure 4.1 (middle) is composed of two windows only).

Candidate Entities with Global Thresholding. The definitions above consider all generated candidates based on local spatial configurations. As the number of candidate entities can be large when visual entities are dense (e.g., big windows with many panels), we additionally impose an upper bound on the number of composite entities considered. The bound, calculated image-wise, is proportional to the number of visual entities, but not larger than a heuristically chosen maximum value on the training set. The candidate generation is done recursively for every image. It starts with a less strict threshold on the `closeBy`/2 relation and it decreases the threshold at each iteration until the constraint on the upper bound of the number of candidates is met. Global thresholding is not necessary for the kernel-based approach. Logical matching, however, even in an approximative form, is computationally more demanding and thus, the distance-based approach requires global thresholding. Another practical alternative could be sampling from the set of candidates.

4.3.2 Visual Interpretation and Problem Definition

Starting from the definition of a logic interpretation, we define a *visual interpretation* of an image at one layer as the union of all relations at that layer.

Definition 7. *A **visual interpretation** is the set of ground entity atoms and the set of ground relationship atoms (over the constant and predicate symbols occurring in the set of clauses that characterize the facade domain) that are extracted from the image and assumed to be true. In a partially observable case, when not all values of all atoms are known, we obtain a **partial interpretation**.*

Visual interpretations of an image at the object and house layers are illustrated in Figure 4.6 and Figure 4.7, respectively. We represent each image as a visual interpretation $I = (x, y)$ or as an instance of a relational database; $x \in \mathcal{X}$ is the set of input ground atoms and $y \in \mathcal{Y}$ the set of output target ground atoms representing the candidates categories. Some of the candidate entities capture the concepts of *window*, *door* or *house*; the rest belong to the category *none*.

Our framework learns from interpretations in a supervised setting [De Raedt, 2008]. We are given a training set of n independent interpretations $D = \{(x_1, y_1), (x_2, y_2),$

$\ldots, (x_n, y_n)\}$. In our problem, the target is the unary relationship `category`/1. Each ground target atom y_i^k in the set of targets y_i belonging to interpretation i, together with its input ground atoms x_i^k, forms a training example $e^k = (x_i^k, y_i^k)$. Each example e^k is, thus, a smaller visual interpretation, part of the larger image interpretation. The goal is to learn a mapping h from the inputs $\mathcal{X}$ to the outputs $\mathcal{Y}$. During prediction, we are given a partial interpretation of an image consisting of ground atoms x, and are required to complete the interpretation using h to predict the output atoms y. The classifier h can have different forms, depending on the machine learning solution employed.

In this work we consider two solutions for composite entity classification. First, we use a k-nearest-neighbor approach based on a distance measure between interpretations. Second, we employ a SVM approach combined with kernels for graphs. We describe these solutions in turn.

4.4 A Relational Distance-based Approach

Following the setup described above, each region of interest example e is represented by its corresponding visual interpretation. We use quantified matchings between two interpretations to build a composite entity classifier using the KNN approach.

Definition 8. *A **matching** between two interpretations I_1 and I_2, denoted $m(I_1, I_2)$, is a mapping such that each atom $\mathbf{a}_1 \in I_1$ corresponds to at most one atom $\mathbf{a}_2 \in I_2$ and vice versa. To each matching we associate a dissimilarity score $d(I_1, I_2)$, which indicates how different the two interpretations are.*

In terms of the graph representation, this corresponds to mapping the vertices from I_1 to those of I_2. A possible matching between two interpretations depicted as graphs, is shown in Fig. 4.8. The quality of the matchings is evaluated by the dissimilarity score. We express this score in terms of a *distance metric* between interpretations. The matching is evaluated directly on the relational language.

4.4.1 The Distance Metric

The *distance function* $d(I_1, I_2)$ that we define measures the quality of the mapping and has two components. One characterizes the *structure* similarity, the other the *appearance*. Our choice is justified by the fact that both aspects may have impact on the matching score.

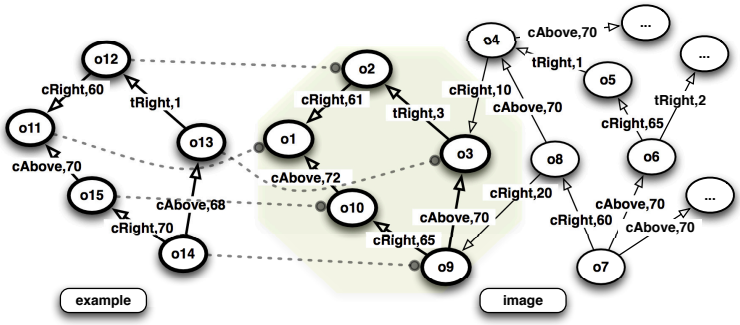


Figure 4.8: Graph representations of an example (left) and an image interpretation (right).

Structural Component

To evaluate how well two logical interpretations match structurally, we must calculate their generalization (or common part). We employ the recent general result of [De Raedt and Ramon, 2009] on metrics, which targets the minimally general generalization of two interpretations, as explained in Section 2.1.2. We choose the object identity ($\mathcal{OI}$)-subsumption order [Ferilli et al., 2003] to calculate it. The minimally general generalization (mgg) is equivalent, in terms of graphs, to the maximum common subgraph. This means that vertices in the subgraph can be mapped to *at most one* vertex in the supergraph, imposing an exact structure matching, and thus the mgg is not necessarily unique [De Raedt, 2008] (Section 2.1.2). Example 17 illustrates the mgg under $\mathcal{OI}$-subsumption for two house facade interpretations.

***Example* 17.** *Let I_1 and I_2 be two visual interpretations, where*
$I_1 = \{\mathtt{part}(\mathtt{o}_1, \mathtt{win}), \mathtt{part}(\mathtt{o}_2, \mathtt{door}), \mathtt{part}(\mathtt{o}_3, \mathtt{win}), \mathtt{cRight}(\mathtt{o}_3, \mathtt{o}_2, 2),$
$\mathtt{cUp}(\mathtt{o}_1, \mathtt{o}_2, 10)\}$ *and* $I_2 = \{\mathtt{part}(\mathtt{o}_4, \mathtt{win}), \mathtt{part}(\mathtt{o}_5, \mathtt{door}), \mathtt{part}(\mathtt{o}_7, \mathtt{door}),$
$\mathtt{cRight}(\mathtt{o}_7, \mathtt{o}_4, 2), \mathtt{cUp}(\mathtt{o}_4, \mathtt{o}_5, 10)\}$.
Under $\mathcal{OI}$-subsumption there are two possible mgg*s:*
$\mathrm{mgg}^0_{\mathcal{OI}}(I_1, I_2) = \{\mathtt{part}(\mathtt{X}_1, \mathtt{win}), \mathtt{part}(\mathtt{X}_2, \mathtt{door}), \mathtt{part}(\mathtt{X}_3, \mathtt{X}_4), \mathtt{cUp}(\mathtt{X}_1, \mathtt{X}_2, 10)\}$
with $\theta^0_1 = \{\mathtt{X}_1/\mathtt{o}_1, \mathtt{X}_2/\mathtt{o}_2, \mathtt{X}_3/\mathtt{o}_3, \mathtt{X}_4/\mathtt{win}\}$, $\theta^0_2 = \{\mathtt{X}_1/\mathtt{o}_4, \mathtt{X}_2/\mathtt{o}_5, \mathtt{X}_3/\mathtt{o}_7, \mathtt{X}_4/\mathtt{door}\}$
$\mathrm{mgg}^1_{\mathcal{OI}}(I_1, I_2) = \{\mathtt{cRight}(\mathtt{X}_1, \mathtt{X}_2, 2), \mathtt{part}(\mathtt{X}_3, \mathtt{X}_4), \mathtt{part}(\mathtt{X}_2, \mathtt{X}_5), \mathtt{part}(\mathtt{X}_1, \mathtt{X}_4)\}$
with $\theta^1_1 = \{\mathtt{X}_1/\mathtt{o}_3, \mathtt{X}_2/\mathtt{o}_2, \mathtt{X}_3/\mathtt{o}_1, \mathtt{X}_4/\mathtt{win}, \mathtt{X}_5/\mathtt{door}\}$,
$\theta^1_2 = \{\mathtt{X}_1/\mathtt{o}_7, \mathtt{X}_2/\mathtt{o}_4, \mathtt{X}_3/\mathtt{o}_5, \mathtt{X}_4/\mathtt{door}, \mathtt{X}_4/\mathtt{win}\}$.

Consequently, the mgg for the two interpretations I_1 and I_2 results in the set $\mathrm{mgg}_{\mathrm{all}} = \{\mathrm{mgg}(I_1, I_2)\}$. Given one mgg in this set $mgg(I_1, I_2) \in \mathrm{mgg}_{\mathrm{all}}$, the

structural distance between two interpretations I_1 and I_2 is:

$$d_s = |I_1| + |I_2| - 2|mgg(I_1, I_2)| \tag{4.1}$$

where $|\cdot|$ is the number of the vertices in the interpretation. From this, it is straightforward to derive a normalized structural distance $d_{ns}(I_1, I_2)$. Similar distance measures are defined in [Nienhuys-Cheng, 1997, Horváth et al., 2001, Kirsten et al., 2000].

Appearance Component

In addition to structural similarities, properties of entities (e.g., color) are important. If mgg represents the maximal common *structure* between two interpretations I_1 and I_2, then $mgg\theta_1$ and $mgg\theta_2$ are specialized maximal common parts of mgg that correspond to I_1 and I_2, respectively. The substitutions θ_1 and θ_2 specify the mapping between different entities. Indeed, if $V/e_1 \in \theta_1$ and $V/e_2 \in \theta_2$ then e_1 is mapped onto e_2. We can now define a *normalized appearance distance* between the two interpretations I_1 and I_2 as:

$$d_{na}(I_1, I_2) = \frac{1}{|mgg|} \times \sum_{a \in mgg} d_0(a\theta_1, a\theta_2),$$

where a is an atom in mgg. Since mgg gives the common structure of the two interpretations, in order to compute $d_{na}(I_1, I_2)$ we start from mgg and specialize each atom $a \in mgg$, such that $a\theta_1$ and $a\theta_2$ are ground atoms with the same predicate symbol a. Let S denote the set of all symbols, then the distance $d_0 : S \times S \to [0, 1]$ is a normalized distance measure defined for our particular application in the following way. Let t_i, s_i be attributes, then:

$$d_0(a(t_1, ...t_n), a(s_1, ..., s_n)) = \frac{1}{n} \times \sum_{i=1}^{n} d_0(t_i, s_i) \tag{4.2}$$

For *discrete* attributes we employ the hamming distance $d_0(t_1, t_2) = 0$ if $t_1 = t_2$, otherwise 1. For *numerical* attributes in the range $[min, max]$:

$$d_0(t_1, t_2) = \frac{abs(t_1 - t_2)}{max - min} \tag{4.3}$$

The Combined Distance

The structural and appearance-based aspects of the distance measure are *combined* into a single measure using a (normalized) weighted average:

$$d_{sa}(I_1, I_2) = w_s \times d_{ns}(I_1, I_2) + w_a \times d_{na}(I_1, I_2), \tag{4.4}$$

where $w_s + w_a = 1$. These weights can be supplied or learned.

Because the mgg of the two interpretations I_1 and I_2 is not unique, the *global normalized distance* between I_1 and I_2 finally is:

$$d(I_1, I_2) = \min_{m \in \text{mgg}_{\text{all}}} d_{sa}(I_1, I_2). \tag{4.5}$$

Next, we employ the kNN classifier. Given the set of candidate entities C in the test image and the set of training examples or prototypes ζ, the algorithm evaluates the quality of each composite entity by computing the distance to the prototypes and classifies it based on the majority vote of its k nearest neighbors. The algorithm returns the set C_{eval} of 3-tuples (y, d, c), where y is the category of $c \in C$ and d is the mean distance score from c to the elements of the subset $\zeta^y \subseteq \zeta$ describing only entities of class y in the set containing the k nearest neighbors.

A strong point of our framework is that distance functions at each level of the hierarchy, either in terms of low-level features or high-level relational spatial composites, can easily be replaced by alternatives. Additionally, it can be proven that the distance that we defined is a metric. This gives extra useful properties, such as the triangle inequality satisfaction, which can be exploited for a faster approximate kNN.

Theorem 1. *The distance function d is a distance metric.*

Proof. d_0 represents the sum of well known distance functions that are metrics. Given that the sum of metrics is also a metric, d_0 is a distance metric. Similarly, d_{na} is an averaged sum of d_0 values and thus, it is a distance metric. d_{sa} is proven to be a metric in [De Raedt and Ramon, 2009]. As a result, their sum d_{sa} is a metric. Finally, a minimum over a set of metrics is also a metric and thus, d is a metric. □

The distance-based approach considers as visual interpretation of a candidate entity the set of visual entities grouped by the candidate, the set of spatial relations amongst them, the set of membership relations, the candidate entity itself and its category relation. Thus, the classification step classifies the candidate entity in a local manner, that is, this step only takes into account the internal structure and appearance properties of the entity to be classified, but no context. This may give unintuitive results at the global level. For instance, it could be that two entities with a significant overlap of two windows are both classified as houses. Although the approach could use, in theory, more visual context at the interpretation level, it is limited by the high computational cost caused by the logical matching, even when an approximation is used in

practice. Therefore, we also perform a *selection* step in which contextual global constraints are taken into account. Using global optimization we find the best subset of the classified entities in the image. From this set we then derive detections. The selection step can be used for the kernel-based approach as well. However, since in practice it is employed only for the distance-based approach, we describe it in this section.

4.4.2 Contextual Candidate Selection

Given the κNN evaluated candidates, the selection step is performed as follows. We first rank the set of candidate entities of interest C_{eval} according to their distances to the nearest examples in ζ. Next, we use a threshold on the number of candidates to select the best set C^*. This step is optional, but recommended as a large space of entities C is usually generated. From this reduced set, we then select those that *together* explain best (most of) the visual entities at that layer. To this end, we formulate the candidate entity selection problem as a *maximum weighted independence set problem.*

Definition 9. *Let $G = (V, E, W)$ be an undirected graph, where V, E and W are the set of vertices and edges and a vertex weighting function, respectively. An **independent set** is a set $S \subseteq V$ such that $\forall e \in E$ the two end vertices of e do not belong to S simultaneously. A **maximum weighted independence set problem** (*WISP*) is formulated as follows: given an input graph $G = (V, E, W)$, find the independence set S of vertices in V such that the value $W(S)$ is maximal.*

In order to convert our problem to a WISP problem we have to find the correspondence to the input graph $G = (V, E, W)$ and the independence set S. In our case:

- V becomes the set of candidate entities C^*.
- we use the set of edges E to model constraints between candidate entities, that is the solution must contain only candidate entities that do not share any visual entities. This constraint is considered through the independence property itself by inserting an edge between any two C which share at least one visual entity:

$$E = \{e(c_1, c_2) | c_1, c_2 \in C^*, V(c_1) \cap V(c_2) \neq \varnothing\}$$

- the vertex weighting function $W : V \rightarrow \mathbb{N}$ is

$$W_c = \sigma(1 - d_\zeta(c, \zeta)), \forall c \in C^*$$

where σ is a function which proportionally amplifies higher scores to ensure the selection of best scored candidate entities. The function that we want to maximize is then $W(S) = \sum_{c \in S} W_c$, where S is one independence set solution.

The solution to the WISP problem is known to be a NP-hard optimization problem. However, both exact and approximation algorithms exist [Busygin, 2006]. For the exact case we use a branch-and-bound algorithm[3] for the maximum clique problem, which is computationally equivalent to the maximum independent set problem computed on the complement graph (for more details see [Östergård, 2002]). For the approximation case we use the algorithm for the maximum weight clique problem proposed in [Busygin, 2006][4]. Other approximation methods are also known to work in polynomial time [Lozin and Milanic, 2010]. However, these are adequate for particular (i.e., planar) graphs, while our selection problem deals with general graphs. If the size of C^* is in a certain range (e.g., ≤ 150 vertices at the object layer) we use the exact optimizer, otherwise the approximate one. This gives acceptable results in practice. The set S^* of selected detections is used next for post processing.

4.5 A Relational Kernel-based Approach with Context

We now include contextual information in a principled way and directly at the interpretation level using the kernel-based approach, which is faster and can evaluate larger example interpretations. Different from the distance-based approach, the kernel-based one works on a graphicalized representation of the visual image interpretation.

As already mentioned, the kernel-based approach is implemented in the kLog framework [Frasconi et al., 2012]. kLog transforms the relational databases into graph-based representations and uses graph kernels to extract the feature space. There are several advantages of using kLog and, implicitly, its kernel-based language. First, it can take relational contextual features into account in a principled and natural way. Its *declarative kernel* allows us to "program" it in order to construct and integrate multiple heterogeneous features via a flexible bias. Second, it allows fast computations with respect to the interpretation size, which allows us to explore different measures of contextual information via the kernel hyper-parameters. Third, kLog provides a flexible architecture in

[3] Available at `http://users.tkk.fi/pat/cliquer.html`.
[4] Available at `http://www.stasbusygin.org`.

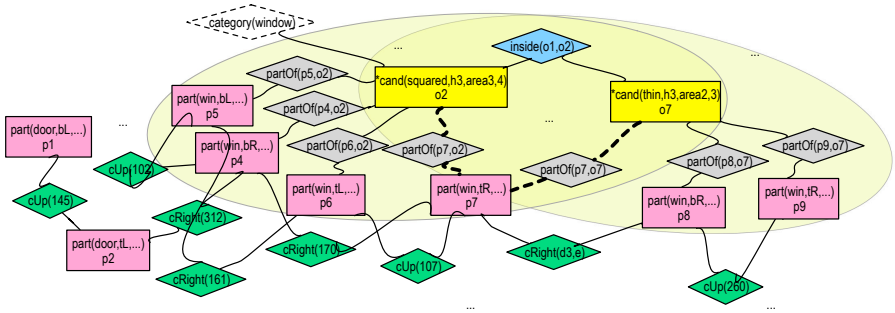


Figure 4.9: Part of the graphicalized visual interpretation in Figure 4.5(a). A neighborhood-pair feature with $R = 2$ and $D = 4$ is marked in yellow. The root vertices or kernel points are the `candidate` vertices and the balls are marked as yellow ellipses.

which only the specification language for relational learning problems is fixed. Actual features are determined by the choice of the graph kernel. In this setting, experimenting with alternative feature spaces is rapid and intuitive.

4.5.1 Graphicalization in kLog

Each interpretation I is converted into a bipartite graph G that has a vertex for each ground atom. Vertices correspond to grounded atoms, either entities or relationships, but identifiers are removed. Edges connect entities and relationships: there is an undirected edge $\{e, r\}$ if the entity identifier in e appears as an argument in r. Figure 4.9 shows part of the graph obtained from the visual interpretation in Figure 4.5(a). Thus, edges connect vertices that share identifiers in the tuples. Role information (i.e., the position of an entity in a relationship) is retained as an edge annotation. The graph can be seen as the result of unrolling (or grounding) the E/R diagram for a particular image. There is no loss of information associated with this step.

4.5.2 The Kernel Function

Once interpretations are represented as graphs, any graph kernel in conjunction with a statistical learner can be used to solve the classification problem in the supervised setting. The kLog implementation uses a variant of the fast

neighborhood subgraph pairwise distance kernel (NSPDK) [Costa and De Grave, 2010]. NSPDK is a decomposition kernel [Haussler, 1999] that counts the number of common parts between two graphs. In our case, the graph is the contextual information of one candidate vertex. Parts are pairs of subgraphs extracted from the graph defined as follows. Given a graph $G = (V, E)$ and a radius $r \in \mathbb{N}$, we denote by $N_r^v(G)$ the subgraph of G rooted in v and induced by the set of vertices $V_r^v \doteq \{x \in V : d^\star(x, v) \leq r\}$, where $d^\star(x, v)$ is the shortest-path distance between x and v. A neighborhood is therefore a topological ball with centre in v. For a given distance $d \in \mathbb{N}$, the *neighborhood-pair* relation is then defined as $R_{r,d} = \{(N_r^v(G), N_r^u(G), G) : d^\star(u, v) = d\}$, where the roots u and v are exactly at distance d. Thus $R_{r,d}$ identifies pairs of topological balls of radius r and distance d. The kernel between two graphs is then the decomposition kernel defined by relations $R_{r,d}$ for $r = 0, \ldots, R$ and $d = 0, \ldots, D$:

$$K(G, G') = \sum_{r=0}^{R} \sum_{d=0}^{D} \sum_{\substack{A,\, B \in R_{r,d}^{-1}(A,\, B,\, G) \\ A',\, B' \in R_{r,d}^{-1}(A',\, B',\, G')}} \kappa((A, B), (A', B')) \tag{4.6}$$

where $R_{r,d}^{-1}(A, B, G)$ returns the set of all pairs of neighborhoods (or balls) (A, B) of radius r with roots at distance d that exist in G. The maximum radius R and the maximum distance D are kernel hyper-parameters and are set experimentally. Figure 4.9 shows a neighborhood-pair feature with $R = 2$ and $D = 4$. The root vertices, also called *kernel points* are imposed via the kLog language as domain bias. In the example, kernel points are the `candidate` vertices and the balls are marked as yellow ellipses. Furthermore we consider a normalized version of $K(G, G')$ that is:

$$K'(G, G') = \frac{K(G, G')}{\sqrt{K(G, G) \cdot K(G', G')}} \tag{4.7}$$

to ensure that relations induced by all values of radii and distances are equally weighted regardless of the size of the induced part sets.

The kernel $\kappa((A, B), (A', B'))$ compares pairs of neighborhood sub-graphs (A, B) and (A', B') extracted from the two graphs G and G' and is defined over sets of vertices (atoms). We ensure that only neighborhoods centered on the same type of vertex will be compared, thus $\kappa((A, B), (A', B'))$ is defined as a product of two components:

$$\kappa((A, B), (A', B')) = \kappa_{root}((A, B), (A', B')) \cdot \kappa_{subgraph}((A, B), (A', B')), \tag{4.8}$$

where the component $\kappa_{root}((A,B),(A',B'))$ is 1 if the neighborhoods to be compared have the same type of roots, while the component $\kappa_{subgraph}((A,B),(A',B'))$ compares the pairs of neighborhood graphs extracted from two graphs G and G'. KLog implements several variants of $\kappa_{subgraph}$ to be used depending on the problem at hand. Because the graphs induced in our application are not sparse and some vertices (those corresponding to visual entities) exhibit large degrees, the likelihood that two neighborhoods match exactly is very low. Thus, we consider a partial or soft match between subgraph pairs. Additionally, we deal both with symbolic and numerical properties and it is more appropriate to solve the classification problem using a *soft* specialization of $\kappa_{subgraph}$ or a *soft match kernel*.

Soft matching

The soft matching kernel uses the idea of multinomial distribution (i.e., histogram) of labels introduced in the Weighted Decomposition Kernel [Menchetti et al., 2005]. Although it discards some structural information inside the graph, context is incorporated by kernel hyper-parameters which collect information from neighboring candidates. More precisely, it counts corners with similar properties in both the instance candidate and the contextually related candidates.

$$\kappa_{subgraph}((A,B),(A',B')) = \sum_{\substack{v \in V(A) \cup V(B) \\ v' \in V(A') \cup V(B')}} \mathbf{1}_{\ell(v)=\ell(v')} \kappa_{tuple}(v,v') \tag{4.9}$$

where $V(A)$ is the set of vertices of A and $\ell(v)$ is the label of vertex v. If the atom $\texttt{part}(p_1, botL, 0.62, door, 0.79)$ is mapped into vertex v, $l(v)$ returns the signature name *part*. In this case $\kappa_{subgraph}$ is decomposed in a part that counts the vertices that share the same labels $\ell(v)$ in the neighborhood pair and ensures matches between tuples with the same signature name, and a second part that takes into account the tuple of property values. In words, we count the vertices that share the same labels in the pair of subgraphs. This is enforced via the first element of the product in the sum, $\mathbf{1}_{\ell(v)=\ell(v')}$, ensuring, thus, matches between tuples with the same signature name. Tuples can have discrete or continuous properties. For the discrete case, the kernel on the tuple considers each element of the tuple independently:

$$\kappa_{tuple}(v,v') = \sum_{d} \delta(prop_d(v), prop_d(v')) \tag{4.10}$$

where $\delta(x, x') = 1$ if $x = x'$ and 0, otherwise. If atom $\texttt{part}(p_1, botL, 0.62, door, 0.79)$ is mapped into vertex v, $prop_d(v)$ returns the property values $botL$ and $door$.

For real values, it considers the standard dot product:

$$\kappa_{tuple}(v, v') = \sum_c prop_c(v) \cdot prop_c(v') \tag{4.11}$$

where for the atom $\texttt{part}(p_1, botL, 0.62, door, 0.79)$, mapped into vertex v, $prop_c(v)$ returns the property values 0.62 and 0.79.

The kernel counts the number of symbolic labels and properties (e.g., `closeRight`, $botL$, $door$, ...) and sums continuous property values that belong to vertices with identical labels $l(v)$ that are contained in the neighborhood pair. For example, in our specific recognition problem at the object layer, the kernel will count, among others, how many corners contained by a candidate and its neighborhood candidates (thus, by a pair of balls) belong to a window, how many belong to a door, how many are of a certain type, how many close to the right relations there are in the neighborhood and so on. For more details, see [Frasconi et al., 2012]. Many alternative statistical learners can be used on the feature vectors created by kLog. In our experiments, we used a standard implementation of support vector machines [Fan et al., 2008], which was integrated via a wrapper in kLog, together with a linear kernel. In practice we train a binary classifier for each category. The set of positively predicted candidate entities is used next for post processing.

4.6 Post Processing

Following the candidate selection step, we employ two post-processing steps.

Bounding Box Prediction

The end goal of our framework is to predict the bounding boxes of detected objects of interest. We use the subgraph of visual entities grouped by the candidate entity to predict a bounding box of the object. This is implemented by mapping the bounding boxes of the visual entities (i.e., vectors of 2D locations) to the corners of the object bounding box.

Non-Maximum Suppression

Either the classifiers or the selection step may give multiple spatially overlapping detections for each instance of an object. The selection step ensures that detections do not share any visual entities, however their bounding boxes may still overlap. Non-Maximum suppression (NMS) however, uses a greedy procedure for eliminating spatially overlapping detections. After applying the bounding box prediction described above we have a set of detections for an object category in an image. Each detection is defined by a bounding box and a score. We sort the detections by score, and greedily select the ones with highest score, while suppressing detections with bounding boxes that are more than some percentage (which varies depending on the setting) covered by a bounding box of a previously selected detection. Similar NMS steps are frequent in the computer vision community (e.g., [Felzenszwalb et al., 2010]). Experimentally, before the evaluation, a NMS step with 50% overlap was applied.

4.7 Experiments

The goal of the experiments, and of this work in general, is not to compete with powerful detectors, often built on dense feature representations, but rather to evaluate how structure and context can be flexibly exploited in understanding house facade images via relational representations. We show that even when starting from relatively sparse cues (primitive layer in Figure 4.10), recognizing and understanding complex objects is feasible, thanks to the use of structure and context. Moreover, rather than just detecting bounding boxes of objects, our methods can return a semantic hierarchical interpretation of the scene, decomposing each object into its constituent parts.

We investigated the following questions experimentally:

(Q1) How does the relational distance (combined with global selection) perform when understanding house facade images? Does the interplay between structural and appearance-based aspects affect the results?

(Q2) How do relational, contextual features perform for images of houses in the kernel-based framework?

(Q3) Which of the two SRL approaches works better?

(Q4) How do the SRL frameworks compare to other propositional approaches?

By answering questions (Q1) we investigate how the relational distance performs in the house facade application. Additionally, we analyse the best parameter settings when understanding house facade images. Question (Q2) looks at the performance of the kernel for the same problem, while using richer contextual features. Finally, the goal of questions (Q3) and (Q4) is to compare the performance results of both SRL frameworks, also in contrast with other propositional approaches.

Notes. As explained in Section 4.4, our relational distance-based approach relies on finding maximum common subgraphs between graphs. This is an optimization problem that is known to be NP-hard. Therefore, in practice, we used an approximation. Additionally, to keep the computation time for one image relatively fast (i.e., in terms of a few minutes), in the case of the distance-based approach, we considered sparser candidate instances by including in the interpretation only visual entities that generate the candidate bounding box (without the inner/sides ones). Moreover, the `partOf`/2 relation was not used in practice (since we do not use contextual candidates, each visual entity is linked directly to the candidate entity via the candidate identifier) and some entity attributes were not used (e.g., corner type confidence). Also for speed reasons, we used as training instances a set of candidate prototypes that were extracted directly from the annotations.

4.7.1 Datasets and Evaluation

For the experimental evaluation we used 2D street view images (Figure 4.1). They commonly display a rich structure (and variety), yet are often quite consistent in terms of structure in a row of houses. We annotated[5] 164 images of rows of house facades from different countries. A number of 20 images were collected by ourselves, the rest from Google Street View. All images showed near-frontal views of the houses and no further rectification was performed. Each image had a resolution of 600x800 pixels. On these images, windows, doors and houses were manually annotated. We considered 2 settings: the full dataset denoted D_{164} and a subset of 60 images denoted D_{60}.

Figure 4.10 recalls the bottom-up data flow. We make use of three layers in a four-level hierarchy: *primitive*, *object* and *house* layers.

The experiments were performed in two different phases. In a first phase, we performed experiments at single layers independently. More precisely, we used as input for the learning task at one single layer the annotated parts at the

[5]Using the LABELME toolbox [Russell et al., 2008].

Figure 4.10: Data flow in the four-level hierarchy of the facades domain. Input layers: pixels, corner primitives and objects. Corresponding output layers: corner primitives, objects and houses, respectively.

house layer and the segmented parts at the object layer. Then, we employed our method to compute the output. In this way, it is possible to get an appreciation of how difficult the learning problem is and what are the limitations of the data at each layer. For the house layer, although the input is precise given the ground-truth annotations, it is still noisy with respect to occlusions of windows and doors present in the image. In a second phase, we performed experiments in the full hierarchical setting, that is, the inputs are image pixels and the outputs are at the house layer. This allowed us to estimate how good the hierarchical approach works.

Because we deal with a *detection* problem we adopt the evaluation measures used in information retrieval. We measure performance in terms of the number of true and false detections in a test dataset. In our setting the positives are all the composite entities selected via the selection function. We evaluate the performance using the overlap measure, which is also the PASCAL VOC [Everingham et al., 2008] criterion. We compare the bounding box BB_d corresponding to the detected concept to the ground-truth bounding box BB_t in manually annotated data. If $area(BB_d \cap BB_t)/area(BB_d \cup BB_t) > 0.5$, then BB_d is a true positive (TP), otherwise it is a false positive (FP). The *precision* P is TP divided by the total number of predicted components. The *recall* R is TP divided by the number of ground-truth components in the test set. The *F1 score* is a measure of accuracy and is the harmonic mean of precision and recall.

The problem of detection is posed in the kernel-based approach as a classification task, namely distinguishing in the image the class of interest with some score. Such a classifier can be turned into a detector by sliding it across the image and

thresholding the scores of the hypotheses to obtain a precision-recall curve. In the distance-based approach, however, our formulation builds on top of a kNN classifier by selecting interesting (already scored) candidates which together find the best semantic segmentation of the image. Since they are together part of the solution, they are all predicted positive instances (except the spatially overlapping ones solved by the final NMS step). As a result, there is no obvious threshold that can be varied to trade-off precision vs. recall and instead of a precision-recall curve, the performance is measured as a precision-recall point.

In the experiments with the relational distance we perform a 5-fold cross validation on the datasets. In the experiments with the kernel we used a fixed split of the data consisting of 48 images for training and 12 for testing. The fixed setup was kept for the comparison experiment in question (Q4). The implementation combines code in Prolog, Matlab and C.

4.7.2 Baselines and Comparisons

To assess the difficulty of the problem we compare our methods to a baseline and to two other well known approaches in computer vision.

Baseline 1. Objectness. As our first baseline we use the generic object detector proposed in [Deselaers and Ferrari, 2010].

Baseline 2. Objectness + HOG. Our second baseline we use the generic object detector as a prior distribution to sample relevant hypotheses in the image. Next, we train[6] a classifier for the category *house* on HOG feature descriptors [Dalal and Triggs, 2005] to re-score them. We first sample 100 house candidates in each image and then employ the specialized classifier to improve the predictions.

Approach 1. Deformable Template Matching with Boosting (DTM). We first employ the boosting approach[7] in [Torralba et al., 2004], which trains an ensemble of weak detectors for each category (*house*, *window*, *door*). Each weak detector computes template matching with a localized patch in object centred coordinates. The features are obtained using a convolution mask tailored to the normalized correlation between the search patch and the deformable template.

[6] Using the LIBSVM library available at: `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

[7] Available at `http://people.csail.mit.edu/torralba/shortCourseRLOC/boosting/boosting.html`.

Individual houses can be more effectively detected using a template matching approach than a texture-based one, since houses in the same row have the same texture and most street scenes greatly vary in texture across the dataset. In our experiments we use 30, 60 and 120 weak classifiers, resulting in the comparison approaches denoted $DTM30$, $DTM60$ and $DTM120$.

Approach 2. Deformable Part-based Models (DPM). The second approach is the deformable part-based models [Felzenszwalb et al., 2010], a system that can represent highly variable objects using mixtures of multiscale deformable part models. Each model is a hierarchical star-structured model defined by a "root" filter (first layer) plus a set of parts filters with spring-like connections between the root and the parts (second layer). The score of a star model at a particular position and scale within an image is the score of the root filter at the given location plus the part scores. The score of a part is the maximum over part filter scores of possible part locations, minus a deformation cost measuring the deviation of the part from its ideal location relative to the root. To discriminatively train this model using object bounding boxes, a latent SVM is used. Results are reported for the standard DPM setting with one component (belonging to the front pose of the house) containing 8 parts. We use as positive examples the house bounding boxes and we provide as "weak" negatives background samples of fixed size from the annotated bounding box surroundings.

4.7.3 Results

We first evaluate the quality of the input primitives introduced in Section 4.2 and illustrated in Figure 4.4, by reporting results at the primitive layer. In this way we can assess the accuracy of the primitives the object layer builds on. We recall that the primitive parts are 2AS interest points with their properties and some of these properties, i.e., relevant part categories, are obtained via two classification steps. For the first classification step establishing whether a corner is relevant or not we obtained $F1 = 0.85$. For the second classification steps distinguishing between window and door corners we obtained $F1 = 0.64$.

(Q1) To investigate the relational distance experimentally, we considered different values of k (in the KNN) at single layers independently and with the full hierarchy on both datasets. The features employed at the object layer rely directly on available detected 2AS from the primitive layer. The corresponding list of relations was {`part`$(idp, type, class)$, `candidate`(idc), `closeUp`$(idp1, idp2, dist)$, `closeRight`$(idp1, idp2, dist)$}. At

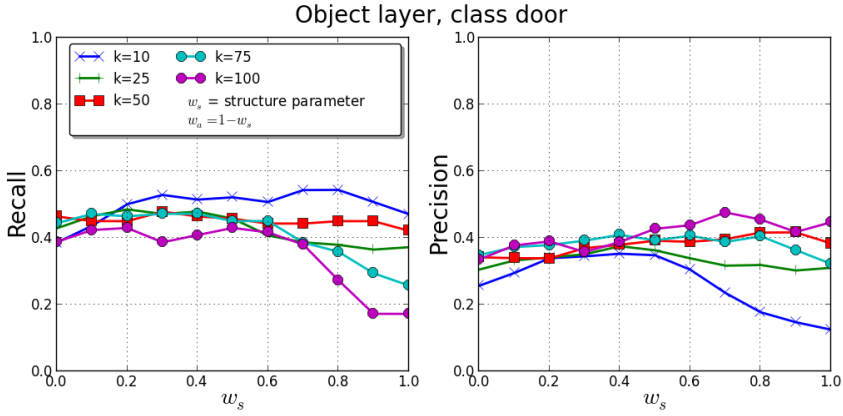


Figure 4.11: Object layer segmentation, class *door*, D_{60}. The influence of the structure component w_s on precision/recall values for different values of k.

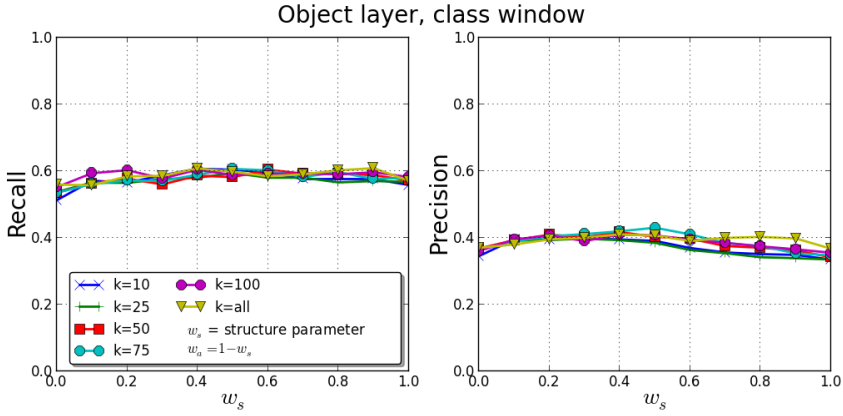


Figure 4.12: Object layer segmentation, class *window*, D_{60}. The influence of the structure parameter w_s on precision/recall values for different values of k.

the house layer, the set of relations used is {`part`$(idp, class)$, `candidate`(idc), `closeUp`$(idp1, idp2, dist)$, `closeRight`$(idp1, idp2, dist)$, `touchRight`$(idp1, idp2, dist)$}.

At the object layer, in the case of D_{60}, the results are shown in Figures 4.11 and 4.12 for classes *door* and *window*, respectively. The maximal values R=0.47,

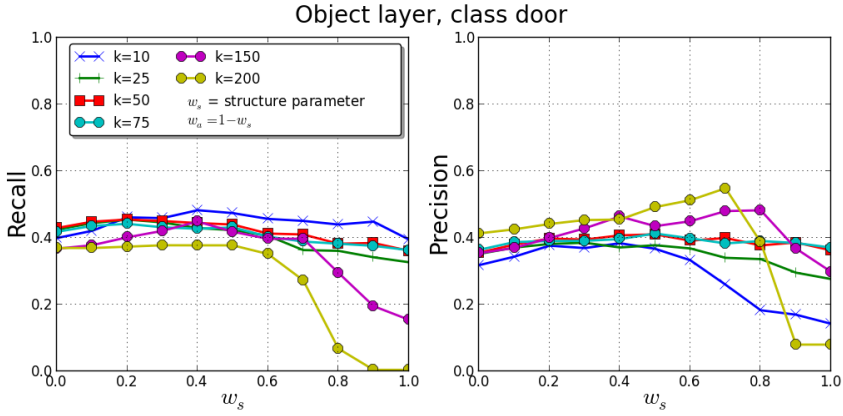


Figure 4.13: Object layer segmentation, class *door*, D_{164}. The influence of the structure parameter w_s on precision/recall values for different values of k.

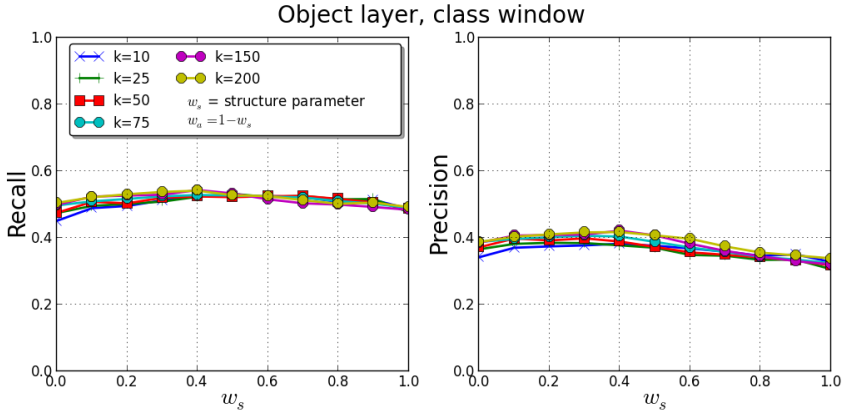


Figure 4.14: Object layer segmentation, class *window*, D_{164}. The influence of the structure parameter w_s on precision/recall values for different values of k.

P=0.41, F1=0.44 for class *door* and R=0.61, P=0.43, F1=0.50 for class *window* were obtained for parameters $k = 75$, $w_s = 0.4$, $w_a = 0.6$ and $k = 75$, $w_s = 0.5$, $w_a = 0.5$, respectively. We notice that for $k = all$ precison and recall for class *door* are very low. That can be explained by the fact that the number of windows is much greater than the number of doors in the training set, and thus, a weighted kNN would be more suitable to try in the future. In the

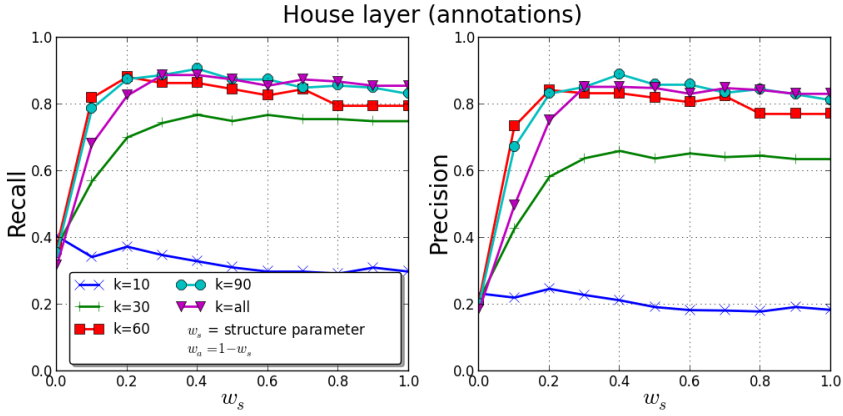


Figure 4.15: House layer segmentation (annotations), class *house*, D_{60}. The influence of the structure parameter w_s on precision/recall values for different values of k.

case of D_{164}, the results are shown in Figures 4.13 and 4.14. The maximal values R=0.44, P=0.41 and F1=0.43 for class *door* and R=0.54, P=0.42 and and F1=0.47 for class *window* were obtained for parameters $k = 50$, $w_s = 0.4$, $w_a = 0.6$, and $k = 150$, $w_s = 0.4$, $w_a = 0.6$, respectively. We notice that results for other k values are also close.

At the house layer, we first tested our approach directly on the ground-truth annotations of the underlying layer, that is, objects such as windows and doors. We observe that if k is large enough ($k \geq 30$ and $k \geq 100$, respectively), increasing of the amount of structure increases precision/recall values. For D_{60}, when $k = 60$, we obtained optimal values R=0.86, P=0.83; for $k = 90$, R=0.92 and P=0.9; for $k = all$, R=0.88 and P=0.85. For D_{164}, when $k = 300$, we obtained optimal values R=0.83, P=0.8 and F1=0.81.

Finally, we evaluated detection results at the house layer using the full hierarchy. From the raw image we first detected the 2AS primitives. These were then employed further as input to detect windows and doors. At this point there are 2 possible ways to proceed. We can select relevant windows and doors via the described selection step at the object layer and use this result as input for the house layer. However, this gave less good results as a high enough recall was required from the object layer to obtain rich enough visual interpretations of images. Alternatively, instead of the full selection step at the object layer, we propagated the top ranked composite entities by directly applying a NMS selection. In this way, the full selection is replaced by a less selective mechanism,

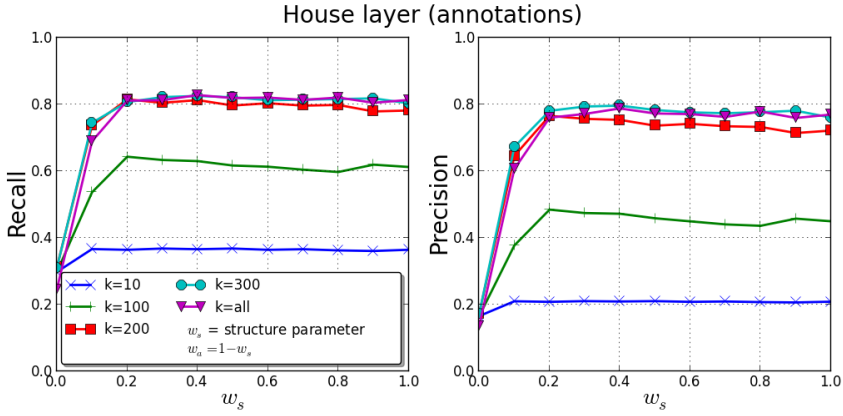


Figure 4.16: House layer segmentation (annotations), class *house*, D_{164}. The influence of the structure parameter w_s on precision/recall values for different values of k.

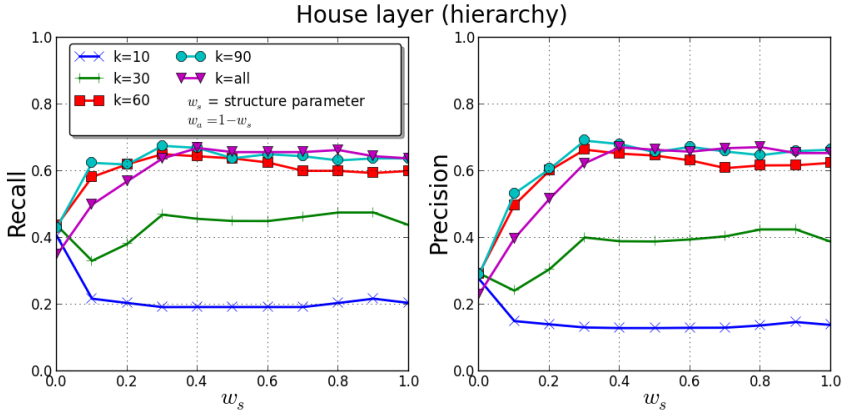


Figure 4.17: Hierarchical segmentation, class *house*, D_{60}. The influence of the structure parameter w_s on precision/recall values for different values of k.

improving recall at the object layer, even when a low NMS overlap degree is used. The selected candidates became visual entities at the house layer. This improved the results to obtain for D_{60}, when $k = 90$, R=0.68, P=0.69, F1=0.68 ($w_s = 0.3$) and when $k = all$, R=0.67, P=0.67, F1=0.67 ($w_s = 0.4$). Figure 4.17 illustrates these results, obtained for a NMS selection with 0% overlap at the

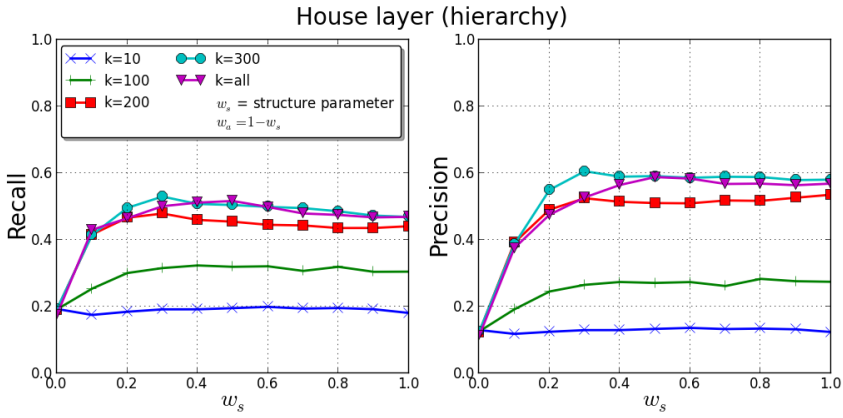


Figure 4.18: Hierarchical segmentation, class *house*, D_{164}. The influence of the structure parameter w_s on precision/recall values for different values of k.

object layer. Results for D_{164} are shown in Figure 4.18. For $k = 300$, we obtained R=0.53, P=0.61, F1=0.57 ($w_s = 0.4$) in the 0% overlap NMS selection setting. When a 2% overlap degree was used to select the best visual entities at the house layer, we obtained R=0.57, P=0.65, F1=0.61 ($w_s = 0.4$) for the class *house*.

We note that, due to the selection procedure, precision and recall are highly coupled in all experiments. Also, our method generalizes well across larger datasets of house facades, independently of the appearance variability. We are able to delineate houses and to separate them from neighboring houses, even when attached. The experiments ran for all settings and their outcome answer the first part of question (Q1). We also indicate that the results reported here are slightly better than those in the papers [Antanas et al., 2012b] and [Antanas et al., 2014]. This is due mainly to the fact that we consider less noisy prototypes which satisfy the 50% overlap criteria with the annotated bounding boxes. This was not the case in our previous experiments where all annotation-derived prototypes were used.

To understand how the interplay between structure and appearance affects the results, we consider the relative weights w_s and w_a (structure vs. appearance for classification) as reference points. More precisely, we evaluated the influence of the structure parameter w_s on precision/recall values.[8] They show that there

[8]We choose w_s as the free parameter; $w_a = 1 - w_s$.

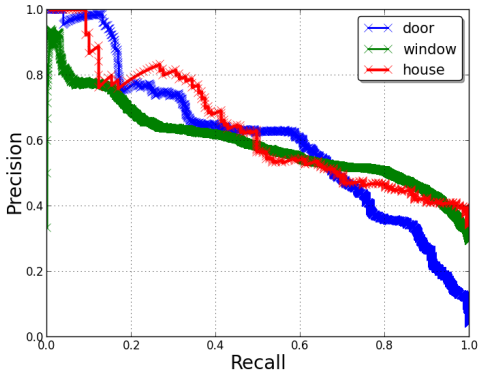


Figure 4.19: PR curves, classes *window*, *door*, *house* using the fixed split, D_{60}, $R = 2$, $D = 4$.

is a "compromise" between the structure and the appearance parameters and that it differs depending on the layer and the setup.

At the object layer, the results reported above show that the variation of the structure has an influence, though it is limited. This can be explained by the fact that windows and doors have mostly the same structure. However, the structure has an indirect influence, as it is needed for computing appearance-based aspects. At the house layer, we varied w_s from 0 to 1 to plot recall and precision. For the annotations setting, these are reported in Figure 4.15 for D_{60} and Figure 4.16 for D_{164}. The best results were generally obtained when $w_s = 0.4$. We stress that w_s is not a threshold to trade precision for recall, but we used it to show the influence of using structure on the performance. Indeed, in our setup the use of structure is essential to obtain good results. We notice that the approach is not very sensitive to a precise value of w_s when $w_s > 0.1$. For the pipeline, the results shown in Figure 4.17 for D_{60} and in Figure 4.18 for D_{164} show an optimal performance for $w_s = 0.3$ (D_{60}) and $w_s = 0.4$ (D_{164}), respectively.

For small values of k recall and precision are much lower for any w_s. This is explained by the fact that, given the structural variability at the house layer, a comparison with enough prototypes is needed. The higher we get in the hierarchy and therefore richer in the semantics, the more relevant the structural aspect becomes.

(Q2) Further, we investigated the use of relational, contextual features for understanding images of houses in the kernel-based framework. For this purpose we trained, in turn, binary classifiers for the classes *window*, *door* and *house* in kLog. At the object layer, we used the following set of features: {`part`$(idp, type, tConf, class, cConf)$, `candidate`$(idc, ar, height, area, noParts)$, `partOf`$(idp, idc)$, `closeUp`$(idp1, idp2, dist)$, `closeRight`$(idp1, idp2, dist)$, `inside`$(idc1, idc2)$, `touchRTaller`$(idc1, idc2)$}. Different from the distance-based approach, because we do not face the computational limitation, we employ denser candidates which include in the interpretation also inner visual entities. Candidates classified as *window* or *door* became parts at the house layer. Here, we used the following set of features: {`part`$(idp, class, score, height)$, `candidate`$(idc, ar, height)$, `partOf`$(idp, idc)$, `closeUp`$(idp1, idp2, dist)$, `touchRight`$(idp1, idp2, dist)$, `inside`$(idc1, idc2)$} for the class *house*.

To assess the impact of contextual features on the performance of each classifier we varied the hyperparameters of the kernel R and D. Figure 4.19 shows precision recall curves for each of the three classifiers. They were obtained for $R = 2$ and $D = 4$, the hyperparameter values that gave the best result. This translates into $F1 = 0.58$ for class *window* and $F1 = 0.54$ for class *door*. The performance obtained for house detection was $R = 0.68$, $P = 0.56$, $F1 = 0.62$. We show that even if we start from relatively sparse cues, the detection problem is solvable with good results thanks to the use of relational representations and kLog's flexible language and kernel. The experiments were performed on the D_{60} dataset using the fixed split.

Many alternative statistical learners can be used on the feature vectors created by kLog. In our experiments, we used a standard implementation of support vector machines [Fan et al., 2008], which was integrated via a wrapper in kLog, together with a linear kernel. The cost c of the SVM was chosen via internal cross-validation on the training set.

(Q3) We compare the relational distance approach to the the graph kernel approach on D_{60} using the fixed split. Comparison results are shown in Table 4.1. We can see that by incorporating more structural context and richer features, kLog improves results over the relational distance. This result is obtained without explicitly considering the global selection step used in the distance-based approach, but directly applying the post-processing.

(Q4) Table 4.2 compares the results of our distance-based approach to those of the baselines on D_{164}. The evaluation setting used is 5-fold cross-validation with the same splits for the distance-based approach and for the baselines. Precision-recall curves are shown in Figure 4.20. Although the baseline detectors

Method	R	P	$F1$
RD (hierarchy) house	0.65	**0.6**	**0.62**
kLog house	**0.68**	0.56	**0.62**
RD window	0.57	0.39	0.46
kLog window	**0.65**	**0.52**	**0.58**
RD door	**0.55**	0.47	0.51
kLog door	0.53	**0.54**	**0.54**

Table 4.1: kLog vs. relational distance (RD); classes *house*, *door* and *window* using the fixed split, D_{60}.

perform reasonably well for the house detection problem, none of these detectors incorporates a fine-grained decomposition of a house, in the form of *structured output* which explains the image in the same way as our relational distance-based framework. DPM is an exception, as the trained model can be visualized in terms of its parts and displacements to the root. Still, these parts do not have an explicit meaning. Moreover, our results are comparable to the DPM results. The relational distance still outperforms the other baselines although we start from sparse features and therefore, a less rich appearance-based component. This is opposed to the employed baselines which are optimized for dense cues and a richer appearance component. The outcome scales to the smaller dataset D_{60} as well. We investigated it by performing similar comparisons to some of the baselines which are reported in Table 4.3. These conclusions implicitly hold for the kernel-based approach as well and therefore, answer question (Q4).

Method	R	P	$F1$
Objectness	0.21	0.08	0.12
Objectness + HOG	0.35	0.10	0.16
DTM 30	0.53	0.55	0.51
DTM 60	0.52	0.51	0.53
DTM 120	0.51	0.54	0.49
DPM	**0.62**	0.61	**0.62**
RD (hierarchy)	0.57	**0.65**	0.61

Table 4.2: Relational distance vs. baselines, class *house*, D_{164} (5-fold cv).

4.8 Related Work

A lot of work in computer vision has focused on fixed compositional structures [Felzenszwalb et al., 2010] or constellation models [Fergus et al., 2007]. Recently,

Method	R	P	$F1$
Objectness + HOG	0.23	0.11	0.14
DTM 120	0.57	0.48	0.52
RD (hierarchy)	**0.68**	**0.69**	**0.68**

Table 4.3: Relational distance (RD) vs. baselines, class *house*, D_{60} (5-fold cv).

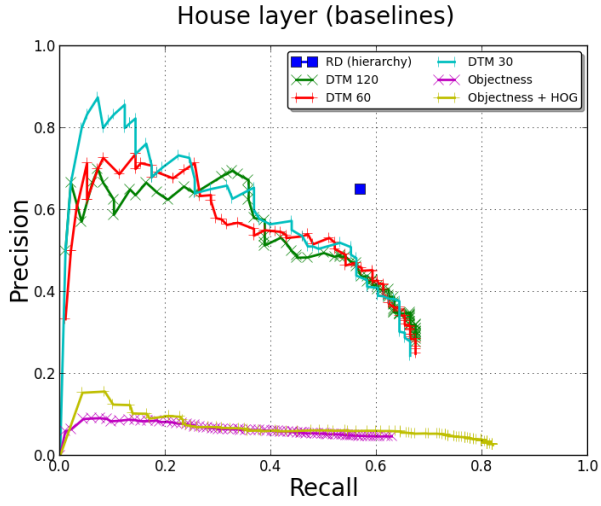


Figure 4.20: Relational distance (RD) vs. baselines. PR curves, class *house*, D_{164} (5-fold cv). We recall that performance for our RD (hierarchy) approach is measured as a precision-recall point due to the selection step.

more attention was devoted to using high-level relational representations for image understanding or object recognition in general. Yet, most of this work is restricted to a model-based approach that performs image interpretation through attributed image grammars [Hartz and Neumann, 2007, Lippow et al., 2008, Han and Zhu, 2009, Felzenszwalb et al., 2010, Girshick et al., 2011, Zhu et al., 2012]. These have been well-studied in the literature [Zhu and Mumford, 2006], but need considerably more input from the user in terms of a set of grammar rules. This is in contrast to our approach, which is based on learning from annotated examples and which uses domain knowledge to specify only basic qualitative spatial relations between image parts.

The use of rich logical formalisms in non-grammar approaches by the state-of-the-art in computer vision is limited [Szeliski, 2010]. Closely related are graph

matching and graph kernel-based techniques for image understanding [Caetano et al., 2009, Harchaoui and Bach, 2007]. However, different from these, our work aims at combining the best of both worlds by using kernel-based and distance-based approaches to learn from logical interpretations. In this sense, our work fits in the current interest for using relational learning techniques for complex vision tasks [Petrou, 2008]. In document analysis, distance-based techniques have been used in a relational setting [Esposito et al., 1992], yet they do not address the intrinsically noisy nature of vision-based interpretation of images of houses. Other relevant work includes approaches based on relational object models [Bar-Hillel and Weinshall, 2008] or probabilistic relational learning [Dubba et al., 2010].

Several papers have addressed the problem of understanding images of house facades. In [Hartz and Neumann, 2007, Hartz, 2009], structure models of meaningful facade concepts are learned from examples, while in [Zhao et al., 2010], the authors tackle the house delineation problem by generating vertical separating lines on the facade and using a dissimilarity measure between these features. Finally, the works in [Müller et al., 2007, Koutsourakis et al., 2009, Terzic et al., 2010, Teboul et al., 2013] assume having the structure or grammar of a building facade and estimate the parameters of the model. In a more advanced setting, [Martinović and Van Gool, 2013] presents an approach to automatically learn two-dimensional attributed stochastic context-free grammars from a set of labeled building facades. Different from these, our work uses distances and kernels between logical interpretations to detect known structures. The relational interpretation in our case lies in the input representation and not in the model itself.

4.9 Conclusions and Future Work

We have presented two new SRL approaches to hierarchically understand images of houses. One approach relies on a new relational distance, while the other is a kernel-based approach nested in kLog, a framework for logical and relational learning with kernels. For the first approach, experiments show the interplay between on one hand structural and appearance-based aspects in the recognition task and on the other hand good detection results both at single layers and at the full hierarchy. Three strong points of the approach are that i) we do not assume availability of a full model of the domain (e.g., a grammar) but only a set of annotated examples, which is more natural and easier to obtain, ii) the framework can easily be extended by adding relational/spatial background knowledge, or replacing the classifiers by other similarity functions or kernels and iii) the approach incorporates a fine-grained decomposition of a house in

the form of structured output which explains the image, as opposed to existing detectors. In the second approach the declarative visual language construction and feature engineering platform used by kLog allows a flexible exploitation of the structural and contextual knowledge in visual scenes. We show that even if we start from sparse cues, our problem is solvable with good results thanks to the use of relational representations and kLog's flexible language and kernel, without the global selection step employed by the relational distance.

This work explores a new relational scheme for solving computer vision problems. Overall, our results clearly show the feasibility and effectiveness of our approach by combining relational knowledge representations with computational vision. One open question for the house facade domain is finding the right semantic representation of the features used as input at the object layer. Here we associate a part to a detected corner. Another possibility is to consider a pair of corners as one part. A clear answer to this question is not obvious and more experimental evaluation is required to settle this issue.

There are several limitations of the frameworks which will be pointed out in the following. First, the declarative features and the visual primitives are tailored to the house facade domain. Although the frameworks can be employed for other domains as well (e.g., human detection), changes with respect to the feature construction are required for good results. Second, the representation relies on strong and discretizable local appearance features and does not guarantee good results for domains in which the appearance component is dominant (e.g., butterflies, trees). Finally, the distance-based approach is limited to deal with rather small interpretations. Once the interpretation size increases, so does the computation time. Nevertheless, the kernel-based approach solves this problem with fast computations on very large interpretations.

4.9.1 Future Work

There are several directions for future work. One direction is going towards a *collective classification* setting, in which target predictions are simultaneously classified based on their features and their dependencies. This is a form of structured output learning in which, not only the input is structured, but also the output. Another direction is to consider hierarchical features as *top-down feedback*. For example, a detected house can constrain the number of doors composing the house, and thus, improve door detection results. A third possibility is to go towards a declarative kernel specification. Such a programmable kernel would allow to fine-tune features of interest for the visual task considered. Although we use the Euclidian distances as continuous attributes, an interesting research direction is building a kernel that integrates

numerical values in a better way than just a simple sum. In this way one can avoid the feature discretization at the primitive layer.

Finally, the declarative definitions of spatial relations are based on general notions of spatial theory. Although they are general and rely on few parameters that are estimated from training data, the integration of a more principled and *richer spatial theory* based on standard frameworks [Cohn and Renz, 2001], would allow a convenient and faster experimental platform for feature construction.

Chapter 5

Relational Scene Classification and Tagging

This chapter continues with a perspective on relational scene understanding which is less domain-focused. We move towards more general scene understanding and show the use of relational representations to understand images belonging to a wide range of indoor and outdoor scene categories. This is an important step towards relational computer vision and closing the loop with the early vision literature. We approach the scene understanding problem via two sub-problems, that is object recognition and scene classification, and show the benefits of relational representations on both of them.

To this end, we contribute by employing a relational language for scene descriptions which builds on automatically detected semantic objects, object parts and spatial relationships that hold among them. Scenes are then described as logical interpretations or, equivalently, as (hyper-) graphs. Using this language, we define qualitative spatial relations, which map object bounding boxes and part locations to higher-order relations among semantic objects and, indirectly, object parts. This mapping is based on domain knowledge in the form of logical rules. When applied for a particular image, the symbolic relations that hold among scene elements characterize their spatial arrangement and provide discriminative cues for the scene/object category. This is a more expressive representation than a fixed grid [Parizi et al., 2012, Lazebnik et al., 2006] and more robust than continuous locations, as it allows to flexibly integrate high-level knowledge about scenes. We show that such relational techniques can also improve scene and object classification. Moreover, our work gives deeper insight in scene understanding by employing higher-order spatial relations

among semantic elements in the scene. The story of the scene is not told only by individual object detections, but also by their complex dependencies.

As explained in Chapter 3, this view on image representation was embraced by early ideas that hierarchical structure and relations are key components of an image understanding system. Differently, however, we start from modern feature and object detectors and more complex visual recognition problems. Consider the images in Figure 5.1. While the pool scene can be distinguished from the others using global information and the office scene from the bar and restaurant categories using the presence of certain objects, both sources of information are prone to mistakes when differentiating between bar and restaurant scenes. In this case, component objects are weaker discriminative cues and it is their complex semantic interaction that helps the scene category disambiguation. Indeed, the differentiating patterns are not the objects themselves but rather the consistent qualitative spatial and functional configurations between chairs. For example, one can describe a bar scene as having "a variable number of chairs of similar size, close to each other and aligned horizontally along a counter". This high-level interpretation of a scene relies on semantically meaningful entities and is consistent across the scene category instances. It can be most generally described using relational representations [De Raedt, 2008] which can naturally capture the alignment relation among the chairs.

Similarly, consistent spatial layouts of objects can be used as contextual information to improve object recognition. Consider, for example, the office scene in Figure 5.1. A possible false detection of a chair north-east to the desk can be discarded by imposing a spatial constraint in the following way: features characterizing chair legs can only be below features characterizing table tops. Also, in this case, we advocate the use of semantically meaningful local features in terms of discrete words. While the inner object appearance and their spatial configuration are characterized by grid-guided histograms of words, such as bag of words (BoW) or spatial pyramids (SP) (see Chapter 2), the qualitative spatial constraints are enforced at the object level between BoW or SP. Additionally, to the presence of other objects in the scene, we consider

Figure 5.1: Sample indoor scenes belonging to categories *inside pool*, *restaurant*, *bar* and *office* (from left to right).

as contextual information the scene category. We approach the scene tagging problem (or object recognition with many object categories) using a similar relational representation of the scene.

This chapter is structured as follows. Section 5.1 presents the visual primitives on which we build the relational representation in Section 5.2. We continue with Section 5.3 which explains how we solve the recognition tasks starting from the relational representation. Section 5.4 shows the benefits of the relational approach experimentally, while Section 5.5 discusses related work. We conclude in Section 5.6.

Part of the work in this chapter was published in:

- Antanas, L., Hoffmann, M., Frasconi, P., Tuytelaars, T., and De Raedt, L. "A relational kernel-based approach to scene classification". In: *Proceedings of Workshop on Applications of Computer Vision*, 2013.

5.1 Scene Primitives

Our relational representation of a scene is built using a set of primitives. A primitive is either an object in the image with its properties, or a global property of the scene. This section describes how we obtain them from raw images. Each scene is characterized by a set of automatically detected objects in the image together with their properties.

Semantic objects First we obtain object detections using off-the-shelf object class detectors introduced in [Felzenszwalb et al., 2010] and [Li-Jia Li and Fei-Fei, 2010] which are freely available online.[1] We do not use all available detectors, but restrict ourselves to a vocabulary of 51 objects, which are more likely to appear in indoor and outdoor scenes. In addition, object classes that characterize very small objects are not considered. Although they may be discriminative for some scene categories (e.g., object `book` for category *office*), they are less likely to be accurately detected by current detectors. In practice, using 51 object detectors is reasonable, given that pre-trained detectors are available. Additionally, we use far less detectors than in [Li-Jia Li and Fei-Fei, 2010]. The set of object classes considered are $\{screen, bed, table, desk, counter, dresser, cupboard, cabinet, mountain, window, bookshelf, people, stair, door, railing, fence, rack, cloth, flower, building, skyscraper, grass, sky, tree, plant, sidewalk, cloud, tower, shelf, mast, ocean,$

[1] Available at `http://vision.stanford.edu/projects/objectbank/` and `http://people.cs.uchicago.edu/~rbg/`, respectively.

$streetlight, soil, flag, cue, pin, sump, drum, boat, bus, bathtub, bridge, beach, horse, cow, animal, sand, streetsign, seashore, truck, rock\}$.

To increase the effectiveness of the detectors, we exploit the idea proposed in [Park et al., 2010], where it is shown that scale-variant, or multiresolution detectors are beneficial to obtain better detections. Thus, we apply the detectors at different scales of the image. If the size of the detected object is small (e.g., bottle), we run the detector at larger image scales, otherwise at smaller ones. Also if the size of the object varies greatly (e.g., car), a larger range of scales is considered. The same number of scales is kept (six to ten depending on the object class) across datasets. We filter out all detections which occupy more than 70% of the image size or less than 1%. Filtered detections at one and three scales are shown in Figure 5.2(a). Next, all detections in the dataset at all scales are globally thresholded keeping the highest scored detections. Even after thresholding, there is a considerable number of false positives as well as missed detections. However, if a detection is a true positive, it is often obtained at many scales and this can be regarded as an implicit detection weight. If a detection is a false positive, its weight is typically much lower. As attributes of each object we use its class label and discretized area.

Object parts In addition to the predicted label, we characterize each object in terms of visual words. We construct a codebook vocabulary on the training images by dense sampling (every 8 pixels) of SIFT features on the bounding boxes of the annotated objects. We then cluster these features to obtain a vocabulary of 1000 words. Given a new SIFT feature, it can be characterized by the closest cluster or word. We then describe each object with BoW or SP, using the built vocabulary.

Global properties In some of our experiments, we also consider global scene properties as primitives. We use the GIST [Oliva and Torralba, 2006, Pandey and Lazebnik, 2011], introduced in Section 2.2.2, and OBJECT BANK [Li-Jia Li and Fei-Fei, 2010] as feature descriptors to globally characterize the whole scene. While the GIST of a scene captures spectral and coarsely localized information across the image [Oliva and Torralba, 2001], the OBJECT BANK descriptor is a collection of scale invariant response maps of many pre-trained object detectors in the scene. Instead of directly using raw descriptors, we use them to separately train individual classifiers on the training instances. The discrete predictions of the classifiers for each image are employed as global scene properties.

Depending on the task, we use different primitives as input to build our relational representation. For the scene classification we consider semantic objects and global properties, while for the object tagging task all three primitives.

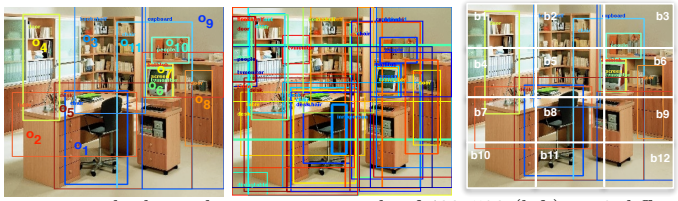
5.2 The Relational Scene Representation

The previous section presented the primitives that we use to describe a visual scene. Next, these are represented as a relational database and serve as input to our *relational language for kernel-based scene understanding*. Practically, the language is specified using the kLog framework [Frasconi et al., 2012], introduced in Chapter 4. Let us now describe how we represent a scene relationally and the tasks considered. We derive the language from its associated E/R data model and thus, is based on entities, relationships linking entities and attributes that describe objects and relationships. The entities in our representations for both tasks correspond to the detected semantic objects with their local properties, as explained in Section 5.1. In Figures 5.3(a) and 5.3(b) these are visually depicted as rectangles. The attributes that describe the objects are represented as ovals. They can be visualized as relational facts, in Figure 5.2(b) for the scene classification problem. For example, the tuple $\texttt{obj}(o_3, \textit{bookshelf}, \textit{large})$ specifies an object entity with o_3 as its identifier. The attributes are its class label (i.e., *bookshelf*) and its area (i.e., *large*). For the scene tagging problem a similar representation can be found in Figure 5.2(c).

Relationships impose a structure on entities and are linked to the entities that participate in the relationships. They are depicted as diamonds. In our representation, we have spatial relationships among objects. These are derived from their spatial bounding boxes. For the scene classification problem, an example is the unary relationship $\texttt{location}(o_1, b_4)$ which associates a specific object $\texttt{o}_1$ with its position on a 3 by 4 rectangular grid that identifies 12 blocks in the original image; see Figure 5.2(a). In particular, for every object O and block B, $\texttt{location}(O, B)$ is true iff the bounding box of O intersects B. The location of the object is conveniently specified with a relationship (and not a property) as the same object can belong to several different locations (blocks) in the image. In a relational representation, we can naturally represent sets of locations that vary in size (and thus with more discriminative power), depending on whether the object is found in a particular block on the grid[2]. Higher order spatial relationships are defined intensionally in Section 5.2.1.

For the scene classification task, other non-spatial relationships are global properties. We encode the global property as a special relationship of zero arity, whose attributes are associated with a scene database. An example is $\texttt{global}(\textit{office})$, where *office* represents the class predicted by a pre-trained classifier (Section 5.1). For the object recognition problem, we encode the visual words characterizing the semantic object as properties of the relations $\texttt{p\_L}_\texttt{i}$ attached to the object entity; i indicates the level in the spatial

[2]This is different from an attribute-based representation, where a fixed vector length is needed.



(a) Office image with object detections at a scale of 400x520 (left), at 3 different scales (middle) and with the spatial grid on top (right).

$x = \{\texttt{global}(office), \texttt{global}(livingroom), \texttt{obj}(o_1, chair, med), \texttt{obj}(o_2, table, med)),$
$\texttt{obj}(o_3, bookshelf, large), \texttt{obj}(o_6, screen, tiny), \texttt{obj}(o_{11}, chair, large), \ldots,$
$\texttt{location}(o_1, b4), \ldots, \texttt{location}(o_1, b11), \ldots, \texttt{location}(o_{11}, b2), \ldots, \texttt{location}(o_{11}, b12),$
$\texttt{aligned\_y}(o_3, o_1), \texttt{aligned\_y}(o_3, o_5), \texttt{aligned\_y}(o_3, o_2), \texttt{aligned\_y}(o_6, o_5),$
$\texttt{aligned\_y}(o_7, o_5), \ldots, \texttt{aligned\_x}(o_4, o_3, o_8), \ldots\}.$
$y = \{\texttt{category}(office)\}.$

(b) Logical visual interpretation of the image for the scene classification task. It shows an example in the scene classification problem. The target attribute is the property (i.e., `office`) of the `category`/1 relation. Each block position B in the relation `location`, is encoded as bi, where i indicates the block on the 3×4 grid.

$x = \{\texttt{obj}(o_1, chair, med), \ldots, \texttt{obj}(o_5, desk, large)), \ldots, \texttt{scene\_category}(o_1, office), \ldots,$
$\texttt{scene\_category}(o_5, office), \ldots, \texttt{p\_L}_0(o_1, w_2), \texttt{p\_L}_0(o_1, w_1), \texttt{p\_L}_0(o_1, w_2), \ldots,$
$\texttt{p\_L}_1(o_1, w_2, b2 \times 2\_11), \texttt{p\_L}_1(o_1, w_2, b2 \times 2\_12), \texttt{p\_L}_2(o_1, w_1, b4 \times 4\_24), \ldots,$
$\texttt{p\_L}_0(o_5, w_3), \texttt{p\_L}_0(o_5, w_1), \texttt{p\_L}_1(o_5, w_1, b2 \times 2\_12), \texttt{p\_L}_2(o_5, w_2, b4 \times 4\_24), \ldots\}.$
$y = \{\texttt{isChair}(o_1), \ldots\}.$

(c) Logical visual interpretation of the image for the object recognition task. It shows examples in the binary *chair* recognition problem. The target is the unary relation `isChair`/1, which is true (and, thus, appears as a relational fact) if the example is positive, otherwise is false (and thus, does not explicitly appear in the interpretation). Each block position B in the relation $\texttt{p\_L}_{k-1}$ is encoded as $bk \times k\_ij$, where k is the size of the grid and ij indicates the block on the grid, $i, j \in \{1, \ldots, k\}$.

Figure 5.2: Visual interpretations of the office scene containing instances of the relational learning tasks considered.

pyramid [Lazebnik et al., 2006]. By having the relation $\texttt{p\_L}_0(O, W)$, we encode that a word W belongs to an object O at level L_0. The relation $\texttt{p\_L}_1(O, W, B)$ associates a word W belonging to an object O with its block position B on a 2 by 2 rectangular grid that identifies 4 blocks in the object patch (i.e., at level L_1). Finally, the relation $\texttt{p\_L}_2(O, W, B)$ does the same thing as $\texttt{p\_L}_1(O, W, B)$, but for a 4 by 4 rectangular grid (i.e., at level L_2). For object recognition the scene category is considered via the unary relation $\texttt{scene\_category}(O, Label)$, where the attribute $Label$ indicates the scene category.

5.2.1 Declarative and Relational Feature Construction

As already explained in Chapter 4, kLog supports extensional relations, explicitly listed sets of facts, and intensional relations, defined within the language using logical rules. In both tasks, the intensional relationships describe spatial arrangements. For the scene classification task, an example is the binary relationship `aligned_y/2` which captures the alignment on the y axis of two objects. Another example is `aligned_x/3` (3 objects aligned on the x axis). These relations are derived using notions of spatial theory and functional properties. For example, `aligned_y/2` is defined for outdoor scenes as a logical rule illustrated in Example 18. It is implemented as a Horn clause in Prolog:

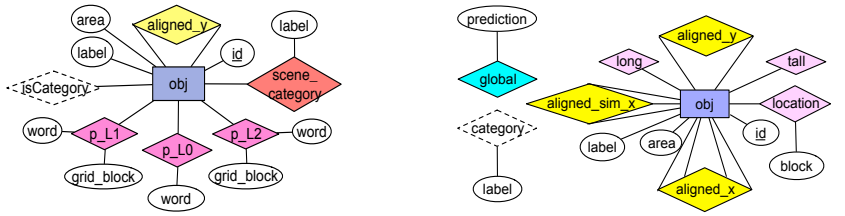
***Example* 18.** *The spatial relation* `aligned_y/2` *is defined as follows:*

$$\texttt{aligned\_y}(O_1, O_2) \leftarrow \texttt{object}(O_1, Label_1, \_), \texttt{object}(O_2, Label_2, \_), \\ \texttt{outdoor}(O_1), \texttt{outdoor}(O_2), \texttt{up}(O_1, O_2).$$

where $\texttt{up}(O_1, O_2)$ is defined based on the bounding boxes of the entities in a similar way. In words, O_1 is above O_2 if the minimum and maximum y coordinates of O_1 are smaller than the minimum and maximum y coordinates of O_2, respectively, and if O_1 is not too much to the right or to the left (in a fuzzy way) of O_2. The relation $\texttt{outdoor}(O_1)$ specifies whether O_1 is an outdoor specific object, and it helps to define the above mentioned relations only between outdoor (respectively indoor) objects. For the object recognition task, a similar spatial relation `aligned_y/2` is defined, although in a slightly changed form. The condition that the objects are typical outdoor objects is removed and the constraint $\texttt{up}(O_1, O_2)$ is replaced by $\texttt{closeUp}(O_1, O_2)$, which indicates that object entities O_1 and O_2, in addition to being vertically aligned, are also spatially close to each other. The closeness is enforced via inferior and superior thresholds on the Euclidian distance between the normalized objects' bounding boxes.

5.3 The Relational Learning Tasks

The ensemble of intensional and extensional relations describing the particular scene gives a *visual interpretation* of an image. It corresponds to a small relational database. Scenes and objects are assumed to be independent for the scene classification task and the object recognition task, respectively. For scene classification, a possible visual interpretation $I = (x, y)$ is shown in Figure 5.2(b), where y is the discrete property *label* of the target relation `category/1` (e.g., $office$). For object recognition, a possible visual interpretation $I = (x, y)$ is



(a) object recognition (b) scene classification

Figure 5.3: E/R modeling of the two tasks. Rectangles denote entity vertices, diamonds denote relationships, and ovals (except obj id) denote properties.

exemplified in Figure 5.2(c), where y is the target unary relation `isCategory`/1 (e.g., `isChair`/1 indicates if the object is a *chair*).

In both tasks we learn from interpretations in a supervised setting [De Raedt, 2008]. Learning to categorize scenes at the relational level is formalized as: given a training set of n independent interpretations $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, the goal is to learn a mapping h from sets of ground atoms $\mathcal{X}$ characterizing scene descriptions to the set of all scene labels $\mathcal{Y}$. We formalize it as a multi-class classification problem. For a new image, we are given its partial interpretation consisting of the input ground atoms x, and are required to complete the interpretation by predicting y using h.

For the object recognition problem each ground target atom y_i^k in the set of targets y_i belonging to interpretation i, together with its input ground atoms x_i^k, forms a training example $e^k = (x_i^k, y_i^k)$. Each example e^k is, thus, a smaller visual interpretation, part of the larger image interpretation. The goal is to learn another mapping h' from the inputs $\mathcal{X} = \{x_i^k\}$ to the outputs $\mathcal{Y} = \{y_i^k\}$. Similarly, during prediction, we are given a partial interpretation of an image consisting of ground atoms x, and are required to complete the interpretation using h' to predict the output atoms y. For each considered object category we train a binary classifier. To solve any of the learning problems, kLog proceeds in three steps: graphicalization, feature generation and the actual learning. The graphicalization step is described in more detail in Chapter 4. Here we focus on the feature generation step for the tasks considered.

5.3.1 Graphicalization in kLog

First, as explained in Chapter 4, each interpretation I is converted into a bipartite graph that has a vertex for each ground relation. Figure 5.7 shows

the graphicalized example corresponding to the scene interpretation in Figure 5.2(b) and obtained using the scene E/R model for scene classification from Figure 5.3(b). Figure 5.4 shows excerpts of graphicalized examples obtained using variants of the E/R model illustrated in Figure 5.3(a) (i.e., without the *area* and *label* properties of the object and the `scene_category`/2 relation). Vertices are annotated by grounded relations and identifiers are removed. The graph can be seen as the result of unrolling (or grounding) the E/R diagram for a particular scene. Again we point out that there is no loss of information associated with this step.

5.3.2 Feature Generation

Once interpretations are represented as graphs, kLog uses a graph kernel in conjunction with a statistical learner in the supervised learning setting. The kernel is a variant of the fast neighborhood subgraph pairwise distance kernel (NSPDK) [Costa and De Grave, 2010], as explained in Chapter 4. Here, we briefly recall some definitions and notations. NSPDK counts the number of common parts between two graphs, where parts are pairs of subgraphs. Given a graph $G = (V, E)$ and a radius $r \in \mathbb{N}$, we denote by $N_r^v(G)$ the subgraph of G rooted in v and induced by the set of vertices $V_r^v \doteq \{x \in V : d^\star(x, v) \leq r\}$, where $d^\star(x, v)$ is the shortest-path distance between x and v. For a given distance $d \in \mathbb{N}$, the *neighborhood-pair* relation is then defined as $R_{r,d} = \{(N_r^v(G), N_r^u(G), G) : d^\star(u, v) = d\}$. The kernel between two graphs is then the decomposition kernel defined by relations $R_{r,d}$ for $r = 0, \ldots, R$ and $d = 0, \ldots, D$,:

$$K(G, G') = \sum_{r=0}^{R} \sum_{d=0}^{D} \sum_{\substack{A,\, B \in R_{r,d}^{-1}(A, B, G) \\ A',\, B' \in R_{r,d}^{-1}(A', B', G')}} \kappa((A, B), (A', B')) \tag{5.1}$$

where $R_{r,d}^{-1}(A, B, G)$ returns the set of all pairs of neighborhoods (or balls) (A, B) of radius r with roots at distance d that exist in G. The maximum radius R and the maximum distance D are kernel hyperparameters. The kernel $\kappa((A, B), (A', B'))$ compares pairs of neighborhood sub-graphs (A, B) and (A', B') extracted from the two graphs G and G' and is defined over sets of vertices (atoms). We ensure that only neighborhoods centered on the same type of vertex will be compared, thus $\kappa((A, B), (A', B'))$ is defined as a product of the two components:

$$\kappa((A, B), (A', B')) = \kappa_{root}((A, B), (A', B')) \cdot \kappa_{subgraph}((A, B), (A', B')), \tag{5.2}$$

where $\kappa_{root}((A,B),(A',B'))$ is 1 if the neighborhoods to be compared have the same type of roots. The component $\kappa_{subgraph}((A,B),(A',B'))$ compares the pairs of neighborhood graphs extracted from two graphs G and G'.

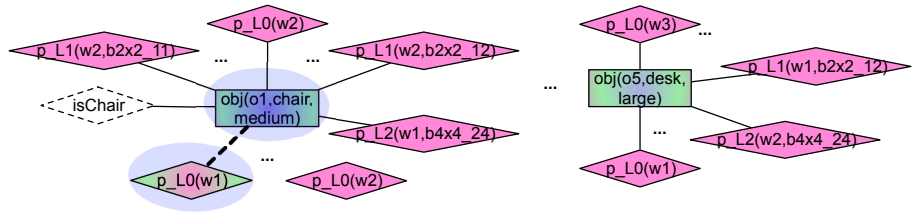
We illustrate and explain how particular kernel variations $\kappa_{subgraph}$ compute meaningful features from the associated graphs of different representations, depending on the task we consider. As shown in Section 5.2, the E/R diagram looks differently depending on the task to be solved. Because of this, the features for the object recognition task will be different from those for scene classification. The E/R scheme also influences the choice of the kernel hyperparameters R and D. We will explain how feature generation works for the two tasks.

Relational Scene Tagging

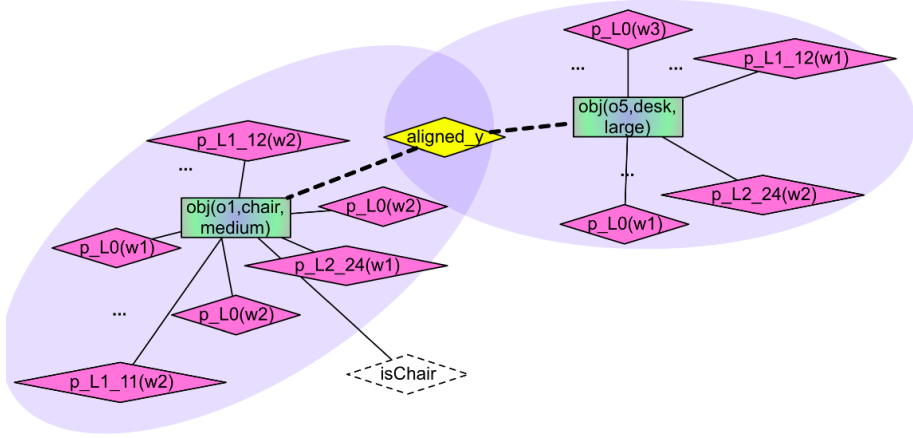
To show the role of relational representations when capturing context information for object recognition, we first consider the setting without any contextual information, and afterwards the setting with context.

Object recognition without context. A partial grounded graph for one scene is illustrated in Figure 5.4(a). One NSPDK feature consisting of a pair of sub-graphs is illustrated in the same figure for $R = 0$, $D = 1$, with kernel roots `obj` and $\texttt{p\_L}_0$. By considering kernel roots $\texttt{p\_L}_1$ and $\texttt{p\_L}_2$ as well, removing the object properties *area* and *label*, and applying exact match on such subgraphs, we can reproduce exactly the SP setting. The exact match kernel is defined as $\kappa_{subgraph}((A,B),(A',B')) = 1$ iff (A,B) and (A',B') are pairs of isomorphic graphs. In words, the overall kernel counts the number of similar graph pairs (or simply graphs) $\texttt{obj} - \texttt{p\_L}_0$, $\texttt{obj} - \texttt{p\_L}_1$ and $\texttt{obj} - \texttt{p\_L}_2$ in the larger graph. Then, it is these counts that are compared when learning. By removing from the E/R model $\texttt{p\_L}_1$ and $\texttt{p\_L}_2$ and keeping `obj` and $\texttt{p\_L}_0$ as kernel roots, we can reproduce with similar hyperparameters the BoW setting.

Evaluating the exact match kernel requires a graph isomorphism subroutine. kLog employs an approximate solution with efficiency guarantees based on topological distances. The idea is to compute an integer pseudo-identifier for each pair of subgraphs, such that isomorphic graphs are guaranteed to have the same identifier and non-isomorphic graphs to have different ones. An identity test between the pseudo-identifiers then evaluates if the graphs are isomorphic or not. The calculation of such identifiers (or features) for one graph is exemplified in Figure 5.5 for the BoW setting. We obtain the identifier by first constructing a topological encoding of each vertex in the graph and then applying a hashing scheme. For the encoding, we relabel a vertex with a sequence that encodes the vertex distance from all other (labeled) vertices (plus the distance from the



(a) No contextual spatial dependencies among objects. The graphs characterizing different object instances are in this case disconnected. The unary relations $\texttt{p\_L}_\texttt{i}$ encode the visual words characterizing the object patch and their grid block locations as properties; i indicates the level in the spatial pyramid.



(b) Spatial context is ensured via the `aligned_y`/2 relation. Grid block locations are in this case included into the signature names, e.g., the relation $\texttt{p\_L}_2(w_1, b4 \times 4\_24)$ becomes $\texttt{p\_L}_2\_24(w_1)$.

Figure 5.4: E/R groundings on a particular image for the object recognition task. Each `obj`/3 relation is a training/testing instance. The target is the dotted diamond. The subgraph pair roots are marked in green. The paths with distances $D = 1$ (case a) and $D = 2$ (case b) are marked with a thick, dashed line. The radiuses $R = 0$ (case a) and $R = 1$ (case b) are marked as ellipses around the roots.

root vertex). For the example in Figure 5.5, the canonical relabeling of the vertex $\texttt{p\_L}_0(w_1)$ is given by the sequence $1root\ 0p\_L_0w1\ 1obj$. This sequence is further hashed into an integer which represents the new label of the vertex. Next, the graph encoding is obtained as the sorted edge list, where each edge is annotated with the endpoints' new labels. The new edge sequence is hashed into the integer pseudo-identifier (i.e., 314 in the example). To obtain the final BoW feature vector that characterizes the (larger) object instance graph, we

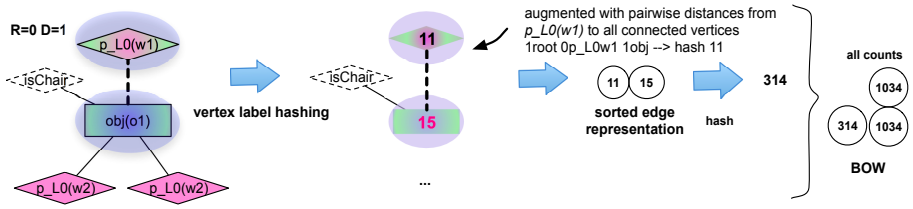
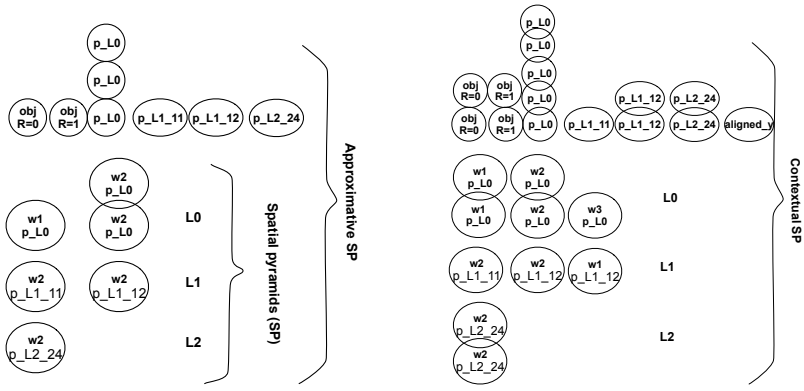


Figure 5.5: Kernel features calculation reproducing the BoW setting. They are obtained using the exact (or hard) match kernel for `obj` and $\texttt{p\_L}_0$ as kernel roots and $R = 0/D = 1$. The graph identifier (i.e., 314) is computed as the hash of the sorted list of edge hashes. An edge hash is computed as the hash of the sequence of the two endpoints new labels (i.e., 11 15). The new label of a vertex is calculated as the hash (i.e., 11) of the sorted list of distance-vertex label pairs (i.e., $1root\ 0p\_L_0w1\ 1obj$).

count the number of identical identifiers or identical pairs $\texttt{obj} - \texttt{p\_L}_0$. For more details about the feature calculation procedure, see [Frasconi et al., 2012].

Object recognition with context. We introduce spatial context for the object recognition task which is realized via spatial constraints between SPs at the object level. Because of the kernel definition (based on pairs of balls) and in order to have fast computations, we slightly change the E/R model to obtain the grounding illustrated in Figure 5.4(b). The difference is the inclusion of the grid blocks into the signature names. More specifically, the relation $\texttt{p\_L}_\texttt{i}$ is replaced by the relation $\texttt{p\_L}_\texttt{i}\texttt{\_jk}$, where jk indicates the block on the grid at level L_i. Thus, in our example, the relation $\texttt{p\_L}_2(w_1, b4 \times 4\texttt{\_}24)$ becomes $\texttt{p\_L}_2\texttt{\_}24(w_1)$. The result is that, if the relation $\texttt{aligned\_y}/2$ is removed and the exact match kernel is replaced by a soft match one with $R = 1/D = 0$, the representation approximates the SP setting. The use of the relation $\texttt{aligned\_y}/2$ in combination with the hyperparameters of the kernel ensures the incorporation of contextual information into the object recognition task.

The soft matching kernel relaxes the idea of all-or-nothing and allows a partial match between subgraphs. To ensure acceptable computational costs, we consider the variant of NSPDK which relies on multinomial distributions (i.e., histogram) of labels. It discards the structural information inside the graph, however, contextual information is incorporated by counting common similarly located words in both the instance object and the contextually related objects.



(a) Kernel features generating approximative SPs.

(b) Kernel features generating contextual approximative SPs.

Figure 5.6: Kernel features calculation for the object recognition task using soft matching without context (a) and with context (b). Hyperparameters used are $R = 1/D = 0$ for (a) and $R = 1/D = 2$ for (b).

The soft match kernel is described in more detail in Section 4.5.2. Here we briefly recall its properties:

$$\kappa_{subgraph}((A,B),(A',B')) = \sum_{\substack{v \in V(A) \cup V(B) \\ v' \in V(A') \cup V(B')}} \mathbf{1}_{\ell(v)=\ell(v')} \kappa_{tuple}(v,v') \tag{5.3}$$

where $V(A)$ is the set of vertices of A and $\ell(v)$ is the label of vertex v. If the atom $\texttt{p\_L}_2\texttt{\_24}(w_1)$ is mapped into vertex v, $l(v)$ returns the signature name $\texttt{p\_L}_2\texttt{\_24}$. In words, we count the vertices that share the same labels in the pair of subgraphs. This is enforced via the first element of the product in the sum $\mathbf{1}_{\ell(v)=\ell(v')}$ ensuring, thus, matches between tuples with the same signature name. The second element $\kappa_{tuple}(v,v')$ deals with the situation when the graphs vertices (relations) have tuples of properties. If these are discrete, we consider each element of the tuple independently:

$$\kappa_{tuple}(v,v') = \sum_{d} 1_{prop_d(v)=prop_d(v')} \tag{5.4}$$

where for an atom $r(c_1,\ldots,c_d,\ldots c_m)$, mapped into vertex v, $prop_d(v)$ returns the property value c_d.

The soft match kernel with hyperparameters $R = 1/D = 0$ and the entity relation `obj`/1 (after removing the object properties `area` and *label*) as kernel root can be used to generate an approximative SP setting. This is a faster alternative to the hard match kernel variant presented in the previous paragraph (which generates the exact SP setting). The advantage of the soft variant is that it can be easily extended to incorporate context in an efficient way. Figure 5.6(a) shows the calculation of an approximative SP using the soft match kernel for the grounding in Figure 5.4(b). The SP feature vector corresponds to the subgraph generated by the entity predicate `obj`(o_1) as its root and the ball with radius $R = 1$ around it.[3] In words, for the feature vector calculation, the kernel counts the number of similar words that are found in similar grid blocks at all levels of the spatial pyramid. A feature is created for each word-block combination. For example, Figure 5.6(a) shows, among other features, one $p\_L_0w_1$ feature and two $p\_L_0w_2$ features. They can be found in graphical form in Figure 5.4(b) belonging to object entity o_1. In addition, the kernel counts the number of times each grid block appears (i.e., three times block p_L_0) and the root itself (i.e., one time $objR = 1$ and one time $objR = 0$).

To incorporate contextual information we change the hyperparameters of the kernel to $R = 1/D = 2$. The calculation of the contextual SP features using the soft match kernel for the grounded graph in Figure 5.4(b) is illustrated in Figure 5.6(b). The kernel considers the words that belong to both object entities which are linked by the vertex `aligned_y`/2. The later is counted as an independent node in the graph and feature in the feature vector. Similarly, the kernel counts, besides the number of times each grid block appears, the number of common words that are in similar blocks, for each grid block and it does not differentiate between grid blocks belonging to different objects.

Relational Scene Classification

For scene classification we use the exact match kernel. One NSPDK feature consisting of a pair of sub-graphs is illustrated in Figure 5.7 for $D = 2, R = 2$. The encoding and canonical labeling of the feature is done in the same way as explained in Section 5.3.2.

For both tasks, many alternative statistical learners can be used on the feature vectors. In the case of scene classification, we used a standard implementation of support vector machines with one-vs-one handling of multiclass classification [Chang and Lin, 2011] integrated via a wrapper in kLog. We chose the cost of the SVM and the set of discriminative relations by performing internal 10 fold cross-validation on the training set. In the case

[3] Thus, it does not include any contextual information.

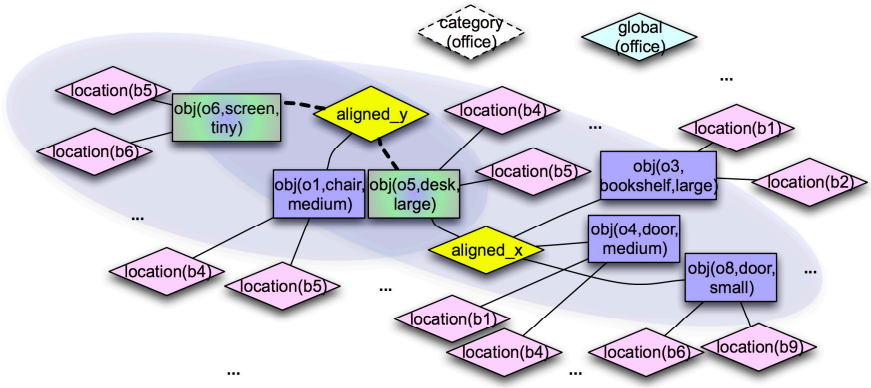


Figure 5.7: Graphicalized (partial) interpretation of the image for the scene classification problem. Illustration of NSPDK features when $D = 2, R = 2$ for the same graphicalized interpretation. The sub-graph pair roots are marked in green. The path with distance $D = 2$ is marked with a dashed line and the radius as ellipses around the roots. The roots are, in this case, nodes with signature name `obj` or object entities.

of object recognition, we used the LIBSVM implementation of support vector machines for binary classification. Since we used the linear kernel, we chose the cost of the SVM by performing internal 5 fold cross-validation on the training set.

5.4 Experiments

Our experiments have two main goals. One goal is to show that combining semantic features with high-level, rich relations between them, may improve scene classification, when we deal with a low semantic inter-category variability. A second goal is to investigate the use of the presence of other object categories as context for the object recognition task. This is similar to the object detection task for several object categories.

This chapter aims at answering the following questions, all in spirit with our mentioned goals:

(Q1) Do symbolic relations among objects improve scene classification?

(Q2) Does the object detection precision influence the classification results?

(Q3) Can relational representations make use of context when tagging scenes with known objects? How helpful is the presence of other objects for this task?

5.4.1 Datasets and Evaluation

In order to evaluate both tasks we consider two datasets: 15Scenes [Lazebnik et al., 2006] and a subset of the 67 MIT indoor dataset which we denote 15MIT [Oliva and Torralba, 2001]. Both of them are used for the scene classification problem, while 15MIT is used for the object recognition task. The 67 MIT dataset contains 67 indoor scene categories and poses a challenging classification problem. The 15MIT considers fifteen categories out of the 67, that are more likely to be confused mostly due to the low semantic inter-category variability. The categories are: {*auditorium, bedroom, computer room, classroom, restaurant, waiting room, inside bus, bar, dining room, concert hall, living room, office, fast-food restaurant, meeting room, kitchen*}. In addition, most of the images have object segmentations together with their object classes, which allows us to use this dataset for tagging scenes with known object categories. The 15Scenes contains six indoor categories {*kitchen, bedroom, living room, meeting room, office, store*} and ten outdoor categories {*suburb, tall building, inside city, industrial, highway, coast, street, open country, forest, mountain*}.

For evaluation we use the same settings as the ones reported in the literature for all datasets. For 15MIT we keep the same training/test split as in [Oliva and Torralba, 2001], where each scene category has about 80 training and 20 test images. For this dataset we do, in addition, a set of experiments in which we only use ground truth instead of object detections in the image, when considering the scene classification task. We consider this experiment to asses the impact on the results when rich and precise information is available. For 15Scenes we keep the same setting as in [Lazebnik et al., 2006]. For both datasets, when employed for the scene classification task, we report the more appropriate overall multi-class prediction accuracy as evaluation measure. As the data for training pairwise category classifiers is balanced, this measure is preferred instead of the average multi-class prediction accuracy (which is more suitable when the data is unbalanced). When considering the object recognition task, we train a binary classifier for each object category using the training/validation data available online. We use the same setting for the 15MIT dataset and use the same fixed split as for scene classification. We report results in terms of average precision (AP) as it is commonly done in the computer vision literature.

5.4.2 Features Used

As we deal with two different tasks, we use different features for each task.

Scene classification

- zero arity relationship `global`. It considers the GIST and OBJECT BANK feature descriptors. We integrate them simultaneously in a discrete way, i.e., as the output of separate classifiers.
- object local properties. The object entity is first assigned as attribute the generic category `object`, such that no identity of the detected object is assumed. Next, the object entity is assigned the object category *label* and its discretized *area*.
- unary relationship `location`. It captures the position of the object on a grid that identifies 3×4 blocks in the image.
- unary relationships `tall`/`long`. If an object bounding box is taller (longer) than 2/3 of the image height (width), then the relation is attached to the object entity.
- higher-order relationships defined on the y axis. The relation `aligned_y`/2 captures the alignment on the y axis of two objects bounding boxes, while `aligned_sim_y`/2 imposes, in addition to the vertical alignment, that the objects must be similar in appearance, i.e., they have the same class labels. Finally, `aligned_sim_y`/3 is similar to `aligned_sim_y`/2, but defined between three objects.
- higher-order relationships defined on the x axis. The relation `aligned_x`/3 captures the alignment on the x axis of three objects. `aligned_sim_x`/3 imposes, in addition, that the objects have the same class labels and `aligned_x`/4 is a quadruple relation that holds between four objects that have similar, but interleaved (e.g., $\mathtt{A} - \mathtt{B} - \mathtt{A} - \mathtt{B}$) object class labels.

In all defined relations, except the unary ones, the objects involved in the relation represent at the same time either indoor, outdoor or natural objects. This condition follows naturally as we do not want to add, for example, a relation between a car and a desk. We refer to all relationships as `all relations` and to local attributes and all relationships, except global properties, as `all`.

Scene tagging

- unary relationships $\mathtt{p\_L_k}$ (or $\mathtt{p\_L_k\_ij}$ variants). They indicate that a visual word characterizes the object detection at level L_k in a spatial pyramid at

	Features	Overall Accuracy (%) 15MIT annot	15MIT	15Scenes
L	object	3.3	4.0	6.7
	labels	33.1	28.4	45.8
R=0/D=0	labels+area	35.5	39.5	64.4
L+R	labels+loc	35.5	34.1	66.7
	labels+tall/long	34.8	35.1	58.7
	labels+area+loc	36.2	46.0	67.6
R=1/D=0	labels+area+loc+tall/long	36.5	46.6	68.3
R=2/D=0	labels+area+all relations	*37.2*	*47.5*	*69.7*
G	Gist	36.6	36.6	52.3
	Object bank	49.7	49.7	80.9
G+L+R	global+all	**60.5**	**54.2**	**82.0**

Table 5.1: Overall accuracy for scene classification on the considered datasets. L denotes local object attributes, R denotes unary/ binary/ ternary/ quadruple relationships and G denotes global information. The best result for the G+L+R setting was obtained when R=2/D=0.

a certain position in the grid. For $k = 0$ the object is characterized by a BoW, while for $k = 2$ the object is characterized by a SP (or approximation of SP). The relationship variant depends on the type of kernel that is declared.

- binary relation `aligned_y/2` among two object detections. It captures the alignment on the y axis of two objects bounding boxes.

Thus, for the reported results we do not consider yet any local properties of the object (such as *area* or *label*) or scene category information. This a first next step to consider for future work.

5.4.3 Results

(Q1) To investigate whether symbolic relations among objects improve scene classification, we analyze the impact of the symbolic information gradually by combining different features. We incrementally incorporate richer and richer relational information to assess the importance of the features used. As a baseline, we use the generic discrete label object as the class of the objects. Next, we replace the object label with the available class label for each detection to see the importance of the semantic features. We add discretized area information (in eight intervals) on the objects and then add gradually functional and spatial

Figure 5.8: Scene images missclassified by OBJECT BANK/GIST and correctly classified by our relational approach (top). Scene images where our approach fails (bottom).

information by incorporating the user defined symbolic relations: `location`, `tall/long` and more complex relations as listed in Section 5.4.2. We report performance results in Table 5.1.

Adding the *area* attribute, unary relations `location` and `tall/long`, separately, improves classification results on both 15MIT and 15SCENES. When they are combined, classification performance increases. We get more improvement when other relations are added. The justification of this result is that when relations between objects are injected in the graph as information about their configuration in the scene, the feature mapping encodes even more high-level, discriminative information about the scene. This increases the classification performance. The hyperparameters of the kernel that gave the best results were R=0 and D=0 for labels+area. For the cases when relations were also used, the best performance was obtained when R=2/R=1 and D=0. Thus, including rich relations and using a larger radius improve classification results, which lets us conclude that indeed symbolic relations are helpful discriminative features. Additionally, we combine global properties `global` with all features. Their combination gave the best results, as shown by the last row of the table.

(Q2) We replace object detections with the ground-truth object annotations and their bounding boxes (15MIT annot) to test if improving the precision of the object detections helps the classification result. Our goal is to show that also when starting from less noisy and rich detectors, relational representations can improve indoor scene classification (see Table 5.1). Again, we gradually include high-level information. We notice that when local semantic information is strong, the impact of qualitative relations is more limited. However, we get

Object category	AP		
	BoW	SP	SP context
chair	32.2	35.6	**36.4**
table	11.5	13.2	**13.8**

Table 5.2: AP for categories *chair* and *table* on the 15MIT dataset. The BoW, SP and SP+context settings were obtained for kernel parameters R=0/D=1, R=1/D=0 and R=1/D=2, respectively.

an improvement on AP of 1.5 using the annotations with the help of relations. This experiment requires fully annotated objects. However, practically, out of the 15 categories considered, only 7 had annotations on the test set and not for all instances. This leads to typically lower accuracy for 15MIT annot than for 15MIT, as Table 5.1 shows.

Qualitative results for the scene classification task are illustrated in Figure 5.8. The bottom row shows mistakes made by our relational approach on the scene classification task. These are mainly due to noisy detections where discriminative qualitative relations between meaningful detections could not be established or where relations between chairs were not properly captured by our spatial theory rules in Prolog. Qualitative relations helped in several cases (top row) by capturing configurations between meaningful detected objects at different image scales (e.g., relations between chairs in the meeting room in the third figure from the left). We note that the complementarity is best visible for 15MIT annot, where the object detections and thus, the relations, are more precise.

(Q3) In order to examine whether relational representations employing context can improve scene tagging, we analyze the performance of recognizing different object categories, by gradually adding local features and contextual information. Similarly, this is done by combining different features. In a first step we consider the setting in which each object detection is characterized by a BoW. Next, we incorporate grid-based spatial information and extend the BoW to a SP. Finally, we consider contextual information for each object by adding the qualitative spatial constraint between SPs.

We perform experiments for categories *chair* and *table* on the 15MIT dataset and the results are reported in Table 5.2. We note that with the SP setting we obtain better results than with the BoW setting, as expected and reported in the literature. By adding contextual information to the SP setting via hyperparameters R=1/D=2, we further improve the performance for both categories.

5.5 Related work

The work presented in this chapter is typically in contrast to current trends in scene classification. They treat a scene as a whole [Oliva and Torralba, 2001], rely on independent semantic objects [Vogel and Schiele, 2007] or use scene parts without spatial information [Zhu et al., 2010] or with weak spatial dependency [Wu and Rehg, 2011, Lazebnik et al., 2006, Zhou et al., 2009], [Yao and Fei-Fei, 2010, Parizi et al., 2012, Quattoni and Torralba, 2009]. These approaches have shown that representations based on objects or parts may provide complementary information to low-level global descriptions and that the semantic configuration of the scene is important. Yet, scenes (indoor in particular) involving many objects that interact in complex semantic patterns, remain a challenge. Recent work [Li et al., 2012, Pandey and Lazebnik, 2011] has shown that stronger spatial cues in the form of geometric constraints between parts can improve results. Still, it uses a part-based model of the scene which captures only pair-wise dependencies between the parts and the root, and which has a fixed number of parts that do not have an explicit semantic meaning. In contrast, we consider qualitative interactions between semantic objects and explicitly describe the image as a logical interpretation.

Relational representations can be used in several ways to solve the scene classification problem. Related work in computer vision using such high-level representations is mostly restricted to grammars [Han and Zhu, 2009, Zhu et al., 2012]. Differently, we employ relational representations in a kernel-based approach that allows us to tractably learn from such complex features. Other related papers that make use of relations between regions of interest [Grauman and Darrell, 2005, Desai et al., 2009, Boutell et al., 2007], or employ kernels for scene classification [Tuytelaars et al., 2011, Harchaoui and Bach, 2007, Grauman and Darrell, 2005, Lazebnik et al., 2006] exist in the literature. However, none of them uses relational representations that build on semantic detected objects together with a kernel-based approach.

5.6 Conclusions and Future Work

The main contribution of this chapter was to show that relational representations are beneficial for more general scene understanding results. To this end, we use global properties, semantic object detections with their local properties, unary relations on the objects and complex spatial relations among them to build a relational database of scene descriptions. This relational database forms the input to a kernel-based statistical relational approach, based on kLog, which models the scene classification and scene tagging problems in a principled and

declarative way. We obtain results competitive with state-of-the-art for scene classification and we show the benefits of relational representations even in a less application-focused setting.

This work gives the perspective of closing the gap between high-level languages, context feedback and low- and mid-level features. It makes a first move towards combining all this information into more general purpose visual systems and broader scene understanding, the main goal of computer vision. However, there are still several limitations of the current approach and implementation due to the assumptions made. First, the representation relies on available, pre-trained object detectors. As a result, if more object categories are required, a detector must first be trained on available datasets. Second, object appearance relies on visual words as mid-level features and is further enhanced with contextual information starting from bag of words and spatial pyramid concepts. Although they work remarkably well in practice for many object categories (as related literature reports), it does not guarantee that visual words and spatial pyramids are the best or most general ways to exploit appearance information for all object categories. Third, there are some limitations of the relational kernel definition and implementation. They become evident when trying to generate desired features in particular settings. For example, in the object recognition problem one could incorporate more effectively contextual appearance information starting from the exact SPs setting and using hard matching. However, due to the language constraints on the kernel roots or, equivalently, the too strict definition of a kernel decomposition "part", desired features are not exactly producible. Currently, the most flexible definition of a decomposition part is a pair of balls centred in kernel roots, with specified radius of balls and distance between the roots.

5.6.1 Future Work

There are many possible directions for future work. An important aspect to investigate next is the effectiveness and performance of the approach on a wider range of object and scene categories. Some categories are better characterized by visual appearance information, while others by the spatial configuration of crucial regions of interest. Ideally, for fast experimentation, this should be done in an integrated framework that disposes of different visual low and mid-level features and forms of representation, i.e., vector form and relational languages. Another imperative line of work is to investigate experimentally the performance of our relational approach on additional object detection benchmarks exhibiting relational contextual information (e.g., Pascal VOC2012 [Everingham et al., 2012]) and to compare it to state-of-the-art results.

Our relational approach relies on the traditional task of learning a single predicate or relation independently. However, classifying multiple objects in the image simultaneously based on their features and dependencies can improve performance. Thus, another interesting direction is to investigate a *collective classification* setting for the scene tagging task. In collective classification the i.i.d. assumption made by the traditional approach to learning no longer holds. Another interesting direction is to investigate how much the scene categorization results are influenced if object detections improve over time. This triggers the reverse question, that is how much the scene category prior can influence scene tagging results. Both tasks can be formulated as a structured output learning problem, which is a more general form of collective classification. In structured output not only the input is structured, but also the output. In this case learning refers to the joint learning of multiple relations simultaneously. To approach it, we would first consider the iterative prediction scheme based on iterative convergence.

Further, more research and experimentation is needed with respect to more object properties (e.g., area or prior category label) and spatial/functional relations. To ease the process, a more convenient and faster experimental platform for feature construction is required. As already mentioned in Chapter 4, achieving this goal for spatial relations can be realized via the integration of a more principled and richer spatial theory implementation based on standard frameworks [Cohn and Renz, 2001] or learning the relations from data.

Finally, a future direction is to investigate other kernels for similar tasks. One interesting possibility is to test the complementarity of the employed relational kernel with other kernels proposed in the literature (e.g., [Rematas et al., 2012]). This can be done by averaging kernel features. A second promising research setup is defining a declarative language for programable kernels to overcome current limitations of kLog, that is a too strict definition of a kernel decomposition "part". The language would allow a more flexible and general definition of a "part" which results in more kernel variants. The user would be able to program more specifically the definition of a decomposition part by relaxing the notion of a pair of balls. For example, one can introduce typed elements that define the decomposition part. The typed part may be defined by a root, a radius and a list of directed relationships filtered by the part. Additional to the typed part, the language could allow a connectivity component that specifies the number of typed parts and the distances between these parts as kernel parameters, indicating the combination of the parts (e.g., pairs, triplets). The typed connectivity could specify a list of directed relationships considered in the combination. The choice of features can be done via the typed part and connectivity elements.

Part II

Relational Recognition for Robot Grasping

Chapter 6

Leveraging World Knowledge and Low-Level Data for Robot Grasping*

Robot vision capabilities are not only essential for perceiving and interpreting the world, but also for acting in arbitrary and dynamic environments. To operate in the real world, a robot also requires good manipulation skills. Objects in the environment can be grasped in different ways. A robot grasp must, at least, satisfy the stability criteria. A good robot grasp depends, in addition, on the specific manipulation scenario: the object, its properties and functionalities, as well as task and grasp constraints (e.g., gripper configuration). How to take into account such information for robot grasping from a high-level and symbolic perspective is a problem that we tackle in this chapter.

Our main contribution is an intermediary *probabilistic logic module for robot grasping which semantically recognizes and reasons about the most likely object category, the performable tasks on the object and the best pre-grasp*, given object properties and potential task constraints. In this context, we define a pre-grasp as a part of the object (e.g., the cup handle) to be grasped. The contribution introduces a symbolic part-based representation, which has the following advantages:

*The work presented in this chapter is joint work with other people on robot grasping. My contribution is the probabilistic logic module which is presented as part of the broader probabilistic logic pipeline.

- grasp transfer to novel objects that share similar parts and thus, generalization over similar (multiple) object parts;
- high-level task-dependent reasoning over parts reduces the space of feasible robot grasps and hence, can improve performance;
- the use of symbolic parts as manifold information leads to reliable object category estimation.

We investigate the role of the probabilistic logic module in robot grasping by leveraging symbolic world knowledge, in the form of object and task ontologies and object-task affordances, object categorical and task-based information. The knowledge and relations are naturally encoded using compact rule-based logical models. However, often, descriptions of the perceived world are also uncertain. For example, not all cups look like the 'prototypical' cup. Thus, we need probabilistic models, which, additionally, allow one to reason about the uncertainty in the world. The module assumes available observations about the task and visual scene and we show that, using our probabilistic logic, we can ask queries about different grasping aspects. By employing object-task affordances and objects/tasks ontologies, the module can generalize over similar object parts and object/task categories when predicting the best graspable object parts (or pre-grasps). This allows us to experiment with a wide range of object and task categories, which is a critical aspect of autonomous agents acting freely in new environments. It is integrated into a *probabilistic logic pipeline*, presented in [Antanas et al., 2013b]. The pipeline also exploits shape features of the object to extract symbolic observations about the visual scene and to execute the grasping. Our approach can be extended beyond the set of categories used, by augmenting the probabilistic logic module with extra rules.

The chapter is structured as follows. Section 6.1 starts by presenting the robot grasping scenario. Next, we explain the task-dependent grasping pipeline in Section 6.2. It briefly presents the vision-based part and focuses on the probabilistic logic module. Before we review related work in Section 6.4, we show experimentally in Section 6.3 the benefits of probabilistic logic reasoning for the robot grasping scenario considered. Finally, Section 6.5 concludes the chapter.

The work on task-dependent grasping pipeline containing the probabilistic logic module has been submitted as:

- Antanas, L., Moreno, P., Figueiredo, R., Neumann, M., Kersting, K., Santos-Victor, J. and De Raedt, L. "High-level Reasoning and Low-level Learning for Grasping: A Probabilistic Logic Pipeline". Submitted to *IEEE Transactions on Robotics*, 2013.

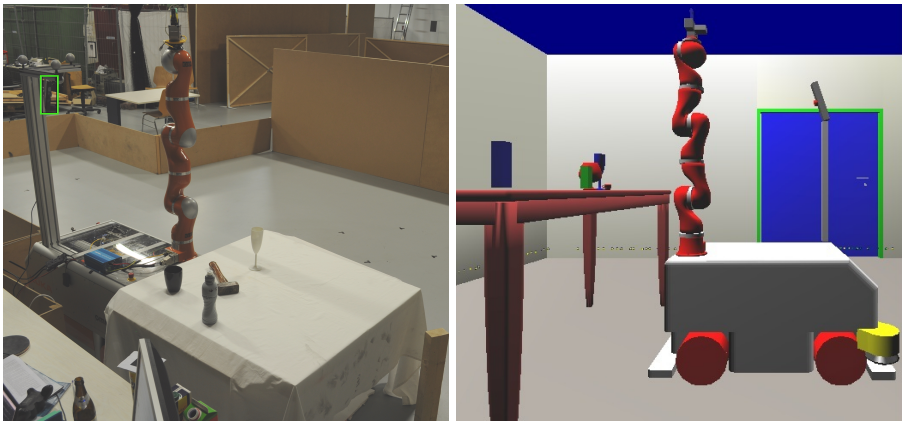


Figure 6.1: Robot grasping scenario. The table is in front of the mobile platform, the arm is vertical, the objects are on the table and the range sensor is marked by the green rectangle.

6.1 The robot grasping scenario

We consider the robot scenario in Figure 6.1. A similar setup is shown with a real robot and with the ORCA robot simulator [Baltzakis, 2005]. The robotic platform is next to a table and on the table there are one or more objects for grasping exploration. The robot has the following components: a mobile component, an arm, a gripper and a range camera (i.e., a Kinect camera in the real robot setup).

An object (e.g., cup, glass) may be placed on the table at various poses. Each pose provides a point cloud, obtained via the Kinect sensor. The points above the table are converted, using segmentation techniques (e.g., [Muja and Ciocarlie, 2012]), into a point cloud describing the object. Figure 6.2 illustrates the partial point cloud of a can placed sideways on the table. The task to be executed on the object (e.g., pour out, pass) by the robot may be given as a scenario constraint. The goal is to determine the best object part to be grasped by the robot, the optimal grasping pose and approach direction of the gripper given the object part, and to execute the grasping.

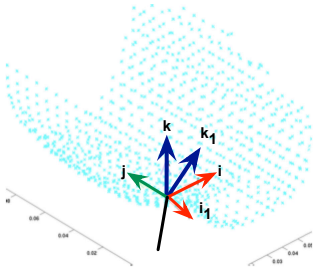


Figure 6.2: A partial point cloud of a can placed on the table. The (i, j, k) is the reference frame of the camera centred at the sample point and its normal is the black line. The (i_1, j, k_1) is the reference frame of the 3D grid, which is obtained by rotating the (i, j, k) frame along the y axis.

6.2 Task-dependent Grasping: A Probabilistic Logic Pipeline

Starting from the defined grasping scenario, the robot first semantically reasons about the most likely object category, task and object part to be grasped using the probabilistic logic pipeline as described below.

6.2.1 The proposed pipeline

The pipeline is exemplified in Figure 6.3. It takes as input the point cloud of an object (e.g., a cup) and, using vision-based techniques, we first obtain a description of the scene in terms of symbolic object parts, object pose and containment. We assess global object similarity via manifold-based graph kernels to complete the scene description with a prior on the object category. Next, using the visual description, we query the probabilistic logic module for the most likely object category, most likely task and best pre-grasp. For our cup example, the manifold shape model predicts the categories *cup*, *can* and *pot* with probabilities 0.56, 0.36 and 0.05, respectively. The categorical logic module reasons about the symbolic parts and recalculates the probabilities as following: 0.98 for category *cup* and 0.02 for category *pan*. The presence of exactly one *handle* increases the probability of the object being more a *cup* rather than a *can* and, as a result, the object is recognized more as a *pan* rather than a *pot*. Similarly, using object-task affordance knowledge and object/task ontologies (e.g., any object affords the tasks *pass* and *pick-place on*), but also world knowledge (e.g., the task *pour in* cannot be executed on a full object), the probabilistic logic module predicts the tasks *pass*, *pick-place on*

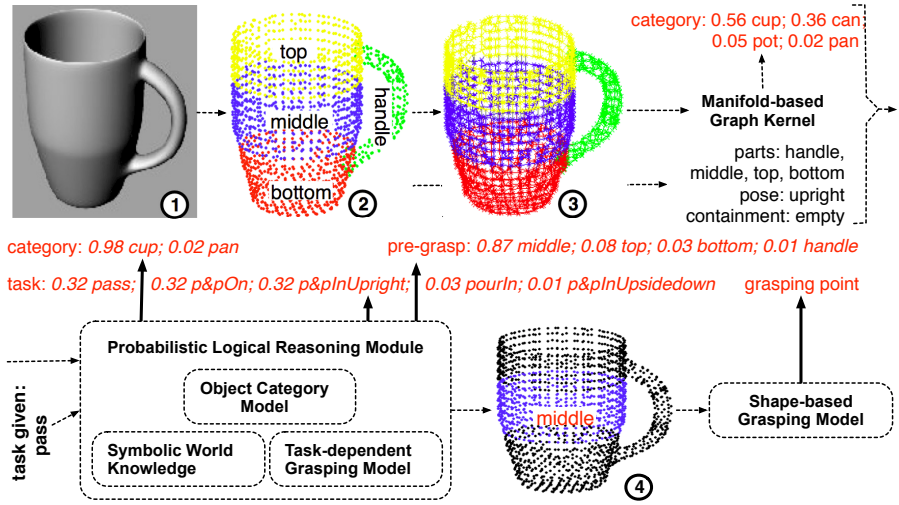


Figure 6.3: The task-dependent grasping pipeline on a *cup* point cloud example. Top row (left to right): object ①, symbolic object parts ② with labels *top* (yellow), *middle* (blue), *bottom* (red), and *handle* (green), k-nn graph ③ with part labels, $k = 4$ (the edges are colored according to the colors of the adjacent nodes), manifolds model with its outcome and visual description of the object (pose, containment and parts). Bottom row: probabilistic logic module with its components and reasoning outcome, predicted pre-grasp *middle* ④, shape-based grasping model and predicted grasping point.

and *pick-place inside upright* with equal probability. If the task given is *pass*, the task-dependent grasping model recognizes as the most likely pre-grasp the *middle* part of the object. The last step in the pipeline is the local shape-based grasping module which predicts the best point for grasping in the pre-grasp point cloud.

An overview of our learning and reasoning grasping pipeline is shown in Figure 6.4. It has four modules. The first module is a visual perception module which maps the object point cloud to scene descriptions in the form of symbolic and probabilistic visual observations about the world. After it segments the object point cloud and performs a full object shape reconstruction, the visual module estimates the object pose (*upright*, *sideways*) and parts (*top*, *middle*, *bottom*, *handle*). Further, it predicts a prior on the object category by employing object similarity based on manifold and semantic part information. The second module is responsible for pre-grasp recognition. It consists of a probabilistic logic reasoning model for task-dependent grasping which, given

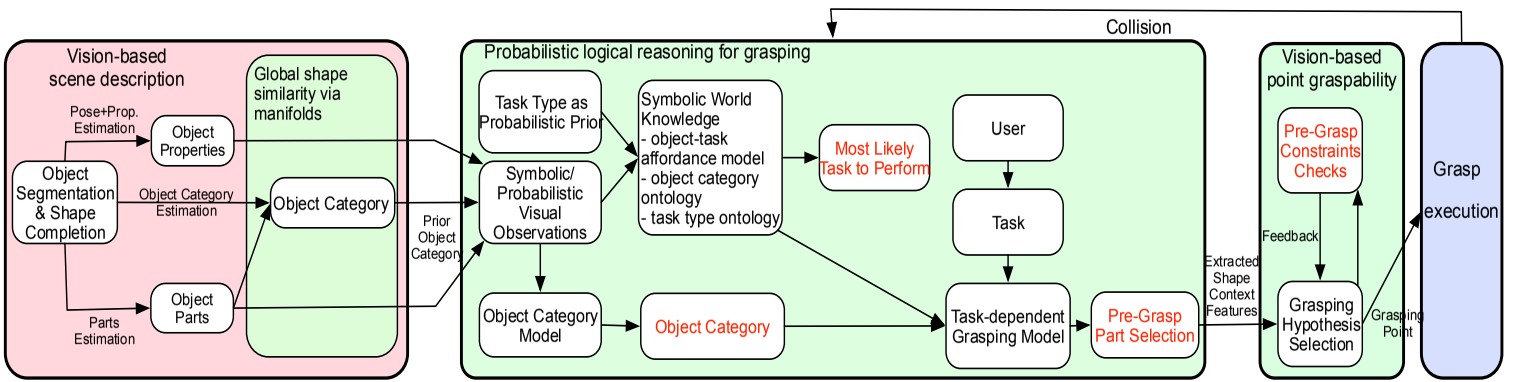


Figure 6.4: The task-dependent grasping pipeline. In blue are marked the vision-based scene module and the grasping execution module. The contributions of this chapter are situated in the green boxes: the probabilistic logic module and the grasping pose prediction module framed in a relational formulation.

the input observations, is able to perform inference about the grasping scenario. The module can answer, in turn, queries about the object category, most likely task and most likely object part to grasp, given the task. It uses probabilistic visual observations about the object, such as object pose, object functionality (i.e., the object is empty or full), and a prior on the object category, and evidence about the task, which is either given or has the form of a prior on the task type. Once we have identified the most likely object part to grasp, the pipeline calls the third module, which solves the problem of grasping pose prediction using visual shape features. Finally, the last module executes the best predicted grasp on the robotic platform. We now describe in turn each of these modules.

6.2.2 Vision-based Scene Description

The role of the visual module (cf. first module box in Figure 6.4) is to obtain a semantic description of the perceived objects in terms of their pose, symbolic parts and probability distributions over possible object categories. The object segmentation step [Muja and Ciocarlie, 2012] is followed by part detection and object category estimation, which rely on a full object point cloud. When only a partial view of the object is available, we first employ a symmetry-based approach for object shape completion. Object reconstruction based on single views is a difficult problem due to lack of observability of the self-occluded parts. Thus, we made some assumptions about the occluded parts, inspired by the work in [Thrun and Wegbreit, 2005, Bohg et al., 2011]. Our shape completion algorithm translates a set of assumptions and rules of thumb observed in many daily environments into a set of heuristics and approximations. They work well for simple box-like and cylinder-like object shapes, such as kitchen-ware tools, and are reasonable assumptions for many other approximately symmetric objects, such as tools (see Figure 6.5). Figure 6.6 illustrates examples of detected semantic parts for several objects using our completion algorithm.

Part-based object representation

We consider two main types of objects: tools and other objects. A tool has as parts a *handle* and a *usable area*, while the rest of the objects have *top*, *middle*, and *bottom* parts and may have *handles*. The extraction of semantical parts is based on the object's dimensions along the main geometrical axes and is achieved by bounding-box analysis. It divides each object into a set of semantical parts, namely, *top*, *middle*, *bottom*, *handle* and *usable area*. When the axis of symmetry is parallel to the supporting plane and the lengths of the remaining directions are smaller than a predefined threshold, we consider that the object has a *handle* and a *usable area*. In order to cope with objects such

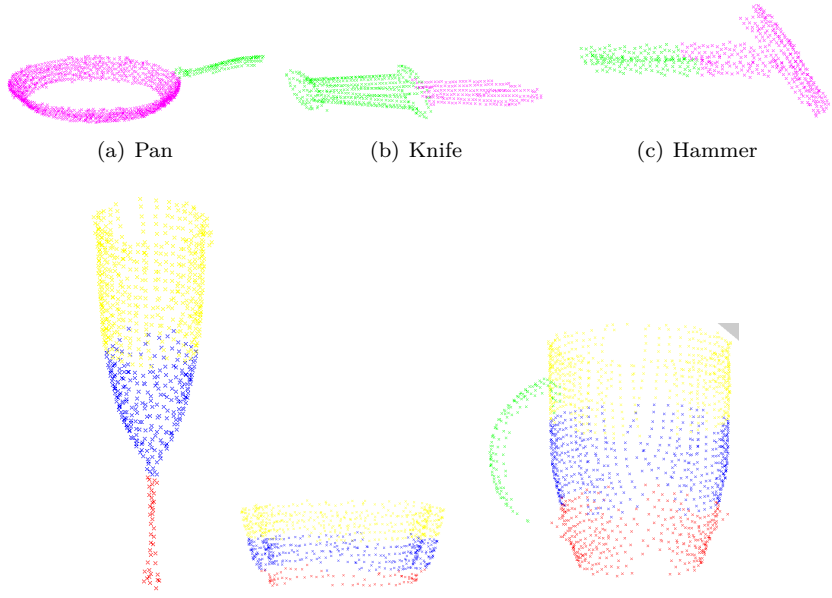
Figure 6.5: Objects having approximative rotational symmetry.

as mugs and pans we detect a handle if a circle is fitted in the projected points with a large confidence. The points lying outside of the circle are labeled as *handle*. The rest of the points are divided along the axis of symmetry into *top*, *middle* and *bottom*. This reduces the search space for robot grasp generation, prediction and planning.

Object category via manifolds

Given the completed object point cloud and its semantic parts, we proceed further at estimating the object category using the object's shape. Global object similarity ensures a strong enough appearance-based predictor for object category. The prediction has the form of a probability distribution over the object categories considered and is used as a prior for the probabilistic logic module. In this way, we leverage low-level and high-level information by combining shape-based global features with semantic logic reasoning. To incorporate global object similarity for object categorization, we leverage propagation kernels, a recently introduced graph kernel designed for classification and retrieval of partially labeled graphs [Neumann et al., 2012b]. We chose the propagation kernel solution as previous work reported in [Neumann et al., 2012a] has successfully employed them to solve the related problem of retrieving similar object views for similar robot grasping datasets. Here we employ them for global object similarity retrieval using the full point cloud instead of single views. The extension is straightforward.

Briefly, we obtain the distribution on object categories for a particular query object by retrieving the objects in an object database being most similar in terms



(a) Pan (b) Knife (c) Hammer

(d) Glass (e) Bowl (f) Cup

Figure 6.6: Semantic parts for several objects after applying the completion algorithm. The colors correspond to parts as follows: yellow - top, blue - middle, red - bottom, green - handle, and magenta - usable area.

of global shape and semantic part information. We represent the objects by labeled graphs, where the labels are the semantic part labels derived by the visual module and the graph structure given by a k-nearest neighbor (k-nn) graph. For each object point cloud we derive a weighted k-nn graph by connecting the k nearest points w.r.t. Euclidean distance in 3D. We use a four-neighborhood (i.e., $k = 4$) and assign an edge weight reflecting the tangent plane orientations of its incident nodes to encode changes in the object surface. The nodes have five semantic classes encoding object part information *top*, *middle*, *bottom*, *handle* and *usable area*. To be able to capture manifold information as graph features in presence of full label information we use a diffusion scheme of the labels corresponding to the diffusion graph kernel, in the following simply referred to as label propagation kernel, as described in [Neumann et al., 2012b]. The graphs of the 3D point clouds as illustrated[†] in Figure 6.3③ capture both manifold

[†]The illustration does not depict the edge weights being proportional to the change in curvature of the adjacent points.

information (geodesic distance) via their structure and semantic information (part labels) via their node labels.

The similarity measure among objects is a kernel function over counts of similar node label distributions per propagation iteration. The T-iteration propagation kernel between two graphs G' and G'' is, then, defined as:

$$K_T(G', G'') = \sum_{t=0}^{T} ker(G'_t, G''_t), \tag{6.1}$$

where T represents the maximum number of label propagation iterations considered and ker is a linear base kernel, defined as:

$$ker(G'_t, G''_t) = \langle \phi(G'_t), \phi(G''_t) \rangle, \tag{6.2}$$

In our experiments we vary $T \in \{0, ..., 15\}$ and use the maximum number of iterations giving the best results.

The main ingredients of propagation kernels are the distribution-based graph features $\phi(G_t)$. They are essentially computed from node label distributions of running label diffusion on the respective graphs. Based on the node label distributions of G_t we compute for each graph the counts of similar distributions among the respective graphs' nodes. As the node label distributions are $m-$dimensional continuous vectors, where m is number of semantic labels (i.e., in this case $m = 5$), propagation kernels use locality sensitive hashing [Datar and Indyk, 2004] as a quantization function to ensure the acquisition of meaningful features (for more details, see [Neumann et al., 2012b]). Propagation kernels leverage the power of evolving continuous distributions as graph features which are built from semantic node labels.

Given a new object G^* that the robot aims to grasp, we first select the top n most similar graphs $\{G^{(1)}, \cdots, G^{(n)}\}$. Then we build a weighted average over the categories of the objects corresponding to $\{G^{(1)}, \cdots, G^{(n)}\}$, which is employed as a prior distribution on the object category for the object with the graph representation G^*. The distribution is further used by the probabilistic logic module to reason about the different grasping tasks. As an example, the prior distribution over object categories for the cup example in Figure 6.3 is 0.56 *cup*, 0.36 *can*, 0.05 *pot*, 0.02 *pan*. We use $n = 10$ in all our experiments. More details about our approach to object categorization and extensive experimental results are given in our paper [Neumann et al., 2013].

6.2.3 The Probabilistic Logic Module

After the visual module, we introduce our reasoning module (cf. second module in Figure 6.4). Its role is to answers three types of queries, related to object category, most affordable task and best semantic pre-grasp. For example, for object category prediction, we query the object instance for being a hammer by calculating the probability $P(cat(O, hammer)|obs, M)$, where O is the object to classify, M is the model and obs is the conjunction of observations made about the world (e.g., $obs = \{\text{parts, pose, task}\}$). Similarly, we query for the most likely pre-grasp by calculating probabilities $P(grasp(Pt)|obs, M)$, where Pt is an object part. When the task is not observed the set of observations becomes $obs = \{\text{parts, pose}\}$ and then the task can be also inferred from the model by calculating probabilities $P(affords(O, T)|obs, M)$, where T is a task.

Bayesian Networks (BNs) are often used to model such complex dependencies involving uncertainty [Madry et al., 2012b]. Differently, our probabilistic model is defined using CP-logic [Vennekens et al., 2009], which has several advantages. First, it can intuitively integrate world knowledge as logic rules. For example, we can exploit object ontologies to reason about object (super-)categories. Similarly, we can use object/task ontologies and task-object affordance models to reason about possible, impossible and desirable pre-grasps. Second, CP-logic is designed to explicitly model causal events (or relationships between random variables). For example, if the object has a usable area and a handle it is likely to be a 'tool' and it can be one (any) of the tool sub-categories (e.g., 'hammer'). This rule is a general piece of information employed locally, as it does not consider other possible causes for the object sub-category. This is rather difficult to encode with a BN, as querying for $P(cat(O, hammer)|obs, M)$ involves knowing all the possible causes for $cat(O, hammer)$ and how they interact with the observations obs. Similarly, if the object is a 'tool' and the task is 'pass', then it should be rather grasped by the usable area instead of by the handle. This involves again local causation. In fact, robotic grasping is characterized by a number of causal uncertain events which sometimes involve different consequences. Third, a CP-theory is more efficient as it requires fewer parameters [Meert et al., 2008] and allows parameter sharing by generalizing over similar situations.

Our grasping CP-theory has 4 parts:

(1) a set of probabilistic observations about the world consisting of visual object properties and, optionally, a probability distribution on the task type;

(2) a set of logic rules, defined as background knowledge which incorporate common sense knowledge about the world, that is, object/task ontologies

and object-task affordances;

(3) a set of probabilistic logic rules as the object category model;

(4) a task-dependent grasping model in the form of probabilistic logic rules.

For all prediction tasks we make the mutually exclusiveness assumption. For object category prediction this implies that an object cannot have several categories at the same time. Similarly, for task selection, this translates into the fact that only one task can be executed at any point in time. We use a ProbLog implementation [Fierens et al., 2011] of the CP-logic theory and we show experimentally that by putting together probabilistic and logical reasoning we improve the grasping performance. We expose the numerated parts in the following subsections.

6.2.4 Observations about the world

We observe one object at a time, however, this setup is easily extendable to consider several objects simultaneously. Visual and task-related observations characterizing the scenario in Figure 6.3 are shown in Example 19. They can be ground probabilistic facts, such as $0.8 :: part(top, o)$ stating that the object o has a top part with probability 0.8, deterministic facts, such as $empty(o)$ stating that o is empty, or CP-rules.

***Example* 19.** *Visual and task-related observations of scenario in Figure 6.3:*
$object(o)$.
$0.8 :: part(top, o)$.
$1.0 :: part(handle, o)$.
$1.0 :: part(middle, o)$.
$1.0 :: part(bottom, o)$.
$0.5 :: pose(o, upright)$.
$empty(o)$.
$0.56 :: cup(o); 0.36 :: can(o); 0.05 :: pot(o); 0.02 :: pan(o) \leftarrow object(o)$.

$pourOut(t_1)$.
$pass(t_2)$.
...
$1/7 :: task(t_1); 1/7 :: task(t_2); ...; 1/7 :: task(t_7) \leftarrow true$.

***Example* 20.** *A CP-rule capturing the prior distribution over the object category for* $object(o)$ *is:*

$0.56 :: cup(o); 0.36 :: can(o); 0.05 :: pot(o); 0.02 :: pan(o) \leftarrow object(o)$.

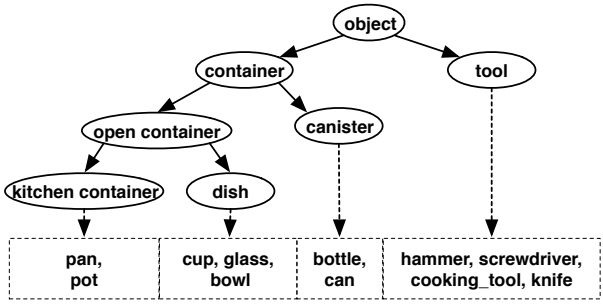


Figure 6.7: An object ontology.

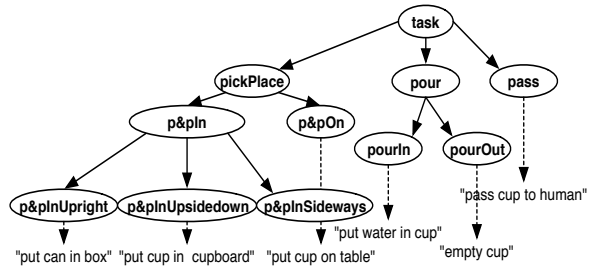


Figure 6.8: A task ontology.

The CP-rule in Example 20 states that an object o belongs to a category with a certain probability, that is, it is either a cup with probability 0.56 or a can with probability 0.36 or a pot with probability 0.05 or a pan with probability 0.02. Similarly, we can have a prior on the task type as a CP-event. In our experiments, if the task is not given, we assume a uniform distribution on the task type. Similarly, if the prior on the object category is not observed, we consider a uniform prior instead.

CP-rules with probability 1.0 are encoded deterministically.

6.2.5 World knowledge: ontologies and affordances

The object ontology is illustrated in Figure 6.7 and structures 11 object categories that we consider in our scenario: $C = \{$*pan, pot, cup, glass, bowl, bottle, can, hammer, knife, screwdriver, cooking_tool*$\}$. The super-categories, defined based on the object functionality, are: *kitchenContainer, dish, openContainer, canister,*

affordances task/object			container							tool			
			open container					canister					
			dish			kitchen							
			cup	glass	bowl	pan	pot	bottle	can	hammer	knife	screwdr	cooking
pass			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
pour		in	✓	✓	✓	✓	✓	-	-	-	-	-	-
		out	✓	✓	✓	-	-	✓	✓	-	-	-	-
p&p	in	upright	✓	✓	✓	✓	✓	✓	✓	-	-	-	-
		upsidedown	✓	✓	✓	-	-	-	-	-	-	-	-
		sideways	-	-	-	-	-	-	-	✓	✓	✓	✓
	on		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 6.1: Object-Task affordances.

container, tool, object. By making use of the ontology structure, the grasping model makes abstraction of the fine-grained object categories.

The task ontology in Figure 6.8 structures 7 tasks: $T = \{$*pass, pourOut, pourIn, p&pInUpright, p&pInUpsidedown, p&pInSideways, p&pOn*$\}$. For example, the task *pourOut* refers to the action of removing the contained liquid, while *p&pInUpsidedown* refers to picking and placing the object inside a shelf in the upside-down pose. Depending on the object properties, its parts and the task to be performed, the object should be grasped in different ways.

The object-task affordances for the considered scenario are illustrated in Table 6.1. They allow us to relate the two concepts and thus define the grasping model in a relational way. Both the affordances table and the object ontology were defined by human experience and inspired by *AfNet: The Affordance Network*, available at: `www.theaffordances.net`. They can be extended to include new object/task categories. This knowledge is translated into deterministic logical rules. For convenience, in the case of task prediction, we defined the affordance rules with a high and equal probability due to the mutually exclusiveness constraint.

***Example* 21.** *We define the world knowledge into rules in the following way:*
%Examples of object ontology mappings
$canister(X) : -can(X).$
$dish(X) : -cup(X).$
$tool(X) : -hammer(X).$
$container(X) : -canister(X).$
$object(X) : -tool(X).$

$object(X) : -container(X).$

%Examples of task ontology mappings
$pour(T) : -pourIn(T).$
$pour(T) : -pourOut(T).$
$task(T) : -pour(T).$

% Task affordances
$possible(X,T) : -object(X), pass(T).$
$possible(X,T) : -container(X), pour(T).$
$possible(X,T) : -object(X), p\&pOn(T).$
$possible(X,T) : -container(X), p\&pIn(T).$
$possible(X,T) : -tool(X), p\&pInSideways(T).$

%Some impossible task affordances
$impossible(X,T) : -canister(X), pourIn(T).$
$impossible(X,T) : -kitchenContainer(X), pourOut(T).$

% Common sense exceptions
$impossible(X,T) : -pan(X), full(X), pass(T).$
$impossible(X,T) : -container(X), full(X), p\&pInUpsidedown(T).$
...
$affords(X,T) : -possible(X,T), not(impossible(X,T)).$

where, for example, $dish(X) : -cup(X)$ is a deterministic intensional rule stating that "any cup is a dish"; $dish(X)$ is the head of the rule, while $cup(X)$ is the body. We initially assume a deterministic affordances model. However, the model can be learned to obtain better estimations [Moldovan et al., 2012] or can be re-estimated in our reasoning module by inferring the most likely task. This shows the flexibility of our approach.

6.2.6 The CP-theory for semantic grasping

Similarly, we use deterministic rules and CP-events indicating object category consequences based on the object parts and properties.

***Example* 22.** *For example the deterministic rule:*
$tool(X) : -part(usable\ area, X), part(handle, X), pose(X, sideways).$

reads as: if the object has a usable area and a handle and it poses sideways, then it is a tool.

When observed parts are, for example, *bottom*, *middle* and *top*, no handles are detected and the pose is sideways or upright, then the object can be a glass, a

bowl or a canister. If the observed pose is upside-down then the object can be a glass, a bowl or a can. If exactly one handle is observed, then the object may be a cup or a pan. In these cases we can define CP-events showing the possible outcomes.

***Example* 23.** *The excerpt of the object categorization theory described above is:*

$0.25 :: glass(X); 0.25 :: bowl(X); 0.5 :: canister(X) \leftarrow part(top, X),$
$\quad part(middle, X), part(bottom, X), no\_handle(X), pose(X, upright).$
$0.25 :: glass(X); 0.25 :: bowl(X); 0.5 :: canister(X) \leftarrow part(top, X),$
$\quad part(middle, X), part(bottom, X), no\_handle(X), pose(X, sideways).$
$0.33 :: glass(X); 0.33 :: bowl(X); 0.33 :: can(X) \leftarrow part(top, X),$
$\quad part(middle, X), part(bottom, X), no\_handle(X), pose(X, upsidedown).$
$0.75 :: cup(X); 0.25 :: pan(X) \leftarrow part(top, X), part(middle, X),$
$\quad part(bottom, X), part(handle, X), pose(X, upright).$

These rules encode generality also by using object super-category atoms in the head. Thus, in order to estimate the object category, we replace the original object ontology defined in Section 6.2.5 with an ontology that models category distributions with respect to the super-categories across the ontology. This is part of the categorization model and is done using CP-events. The causal probabilities are estimated based on the number of specific categories in the leafs. For example, we have the distribution over *hammer*, *knife*, *screwdriver* and *kitchen_tool* caused by the super-category *tool* or the distribution over *can* and *bottle* caused by the object being a *canister*:

***Example* 24.** *An excerpt of the object ontology that models category distributions w.r.t. the super-categories is:*

$0.25 :: hammer(X); 0.25 :: knife(X); 0.25 :: screwdriver(X);$
$\quad 0.25 :: kitchen\_tool(X) \leftarrow tool(X).$
$0.5 :: can(X); 0.5 :: bottle(X) \leftarrow canister(X).$

Thus, in our experiments, the object categorization CP-theory M contains category rules, world knowledge and visual observations. As explained in Chapter 2, interpreting the second rule in the example as $P(can(o)|canister(o)) = 0.5$ is incorrect. The conditional probability that o is a can, given that o is a canister, may be different than 0.5 if there is a second cause that contributes to $P(can(o))$.

Querying for the most likely object category is equivalent to calculating $argmax_C P(cat(o, C)|M)$. It is possible to ask the query without grounding the specific object category. This will result in a probability distribution over

object categories, which is better than the observed prior. For the example in Figure 6.3 the new distribution is $P(cat(o, cup)) = 0.98, P(cat(o, pan)) = 0.02$, while $P(cat(o, can))$ and $P(cat(o, pot))$ become 0.

There are different levels of generalization with respect to the rules of the theory. We experimented also with more general rules to investigate the suitability of our model. A more general theory was also able to improve the object category prior (see section 6.3), showing similar behavior and results to the more specific one. Examples of more general rules are shown below, where we replace, for example, the more specific head *0.25::glass(X); 0.25::bowl(X); 0.5::canister(X)* with the super-category *1.0::container(X)*, while keeping the same rule body. For the example in Figure 6.3 the new distribution with the more general theory becomes *P(cat(o,cup))=0.93, P(cat(o,pot))=0.05, P(cat(o,pan))=0.02.*

***Example* 25.** *The excerpt of the more general object categorization theory is:*

$1.0 :: container(X) \leftarrow part(middle, X), part(top, X), part(bottom, X),$
$no\_handle(X), pose(X, upright).$
$0.6 :: dish(X); 0.4 :: canister(X) \leftarrow part(top, X), part(middle, X),$
$part(bottom, X), no\_handle(X), pose(X, sideways).$
$1.0 :: container(X) \leftarrow part(middle, X), part(top, X), part(bottom, X),$
$no\_handle(X), pose(X, upsidedown).$
$0.5 :: cup(X); 0.5 :: kitchen\_container(X) \leftarrow part(middle, X), part(top, X),$
$part(bottom, X), one\_handle(X).$

To *query for the most likely task* in a grasping scenario, we use world observations, the probability distribution over object categories and world knowledge. We can estimate the most likely task by calculating $argmax_T P(affords(o, T)|M)$, where M is the CP-theory. Again, one can ask queries without grounding T to obtain a probability distribution over possible task types. For the example in Figure 6.3 the distribution over possible tasks is $P(affords(o, pass)) = 0.32$, $P(affords(o, p\&pOn)) = 0.32$, $P(affords(o, p\&pInUpright)) = 0.32$, $P(affords(o, pourIn)) = 0.03$, $P(affords(o, p\&pInUpsidedown)) = 0.01$.

We define the pre-grasp recognition model as a set of CP-events. Each causal event generates as consequence the graspability of a certain object part conditioned on the part existence, task, object (super-)category and properties. The feasibility of the semantic grasp is encoded via the causal probability. Some examples from the grasping model for the *dish* super-category are shown in Example 26:

***Example* 26.** $0.8 :: grasp(X, T, middle) \leftarrow affords(X, T), pass(T), dish(X),$
$pose(X, upright), full(X), part(middle, X).$
$0.1 :: grasp(X, T, top) \leftarrow affords(X, T), pass(T), dish(X), pose(X, upright),$
$full(X), part(top, X).$

$0.1 :: grasp(X, T, bottom) \leftarrow affords(X, T), pass(T), dish(X), pose(X, upright),$
$\quad empty(X), part(bottom, X).$
$0.6 :: grasp(X, T, middle) \leftarrow affords(X, T), pass(T), dish(X), pose(X, upright),$
$\quad empty(X), part(middle, X).$
$0.2 :: grasp(X, T, top) \leftarrow affords(X, T), pass(T), dish(X), pose(X, upright),$
$\quad empty(X), part(top, X).$
$0.1 :: grasp(X, T, handle) \leftarrow affords(X, T), pass(T), dish(X), pose(X, upright),$
$\quad full(X), part(handle, X).$
$0.2 :: grasp(X, T, bottom) \leftarrow affords(X, T), pass(T), dish(X),$
$\quad pose(X, upsidedown), part(bottom, X).$
$0.7 :: grasp(X, T, middle) \leftarrow affords(X, T), pass(T), dish(X),$
$\quad pose(X, upsidedown), part(middle, X).$
$0.1 :: grasp(X, T, top) \leftarrow affords(X, T), pass(T), dish(X),$
$\quad pose(X, upsidedown), part(top, X).$
...
$0.7 :: grasp(X, T, middle) \leftarrow affords(X, T), p\&pIn(T), dish(X),$
$\quad pose(X, sideways), part(middle, X).$
$0.3 :: grasp(X, T, bottom) \leftarrow affords(X, T), p\&pIn(T), dish(X),$
$\quad pose(X, sideways), part(bottom, X).$
$1.0 :: grasp(X, T, middle) \leftarrow affords(X, T), pourIn(T), dish(X),$
$\quad empty(X), part(middle, X).$
$1.0 :: grasp(X, T, middle) \leftarrow affords(X, T), pourOut(T), dish(X),$
$\quad not(empty(X)), part(middle, X).$
...

We can enforce constraints to model impossible pre-grasps. For example, when the object is a *tool* and the task is *pour*, we have an impossible affordance and thus, an impossible pre-grasp position. In the same way we could specify and integrate constraints from the motion planner to model impossible pre-grasps due to collision.

***Example* 27.** *Examples of constraint rules in ProbLog are:*

%Constraint for impossible affordances
$false : -grasp(X, T, R), task(T), object(X), impossible(X, T), part(R, X).$
%Constraint for collision
$false : -grasp(X, T, R), task(T), object(X), part(R, X), collision(R).$
%Other constraints
$false : -grasp(X, T, R), pose(X, upsidedown), pan(X), task(T), part(R, X).$
...

The first constraint states that it is impossible that the pre-grasp atom $grasp(X, T, R)$ is true when the body is true. This will guarantee that the

probability of such grasps is equal to 0. The second constraint rule shows that, additionally, we can connect the reasoning module to the execution planner by enforcing the probability of a pre-grasp to 0 if there are environmental constraints for the gripper. The third constraint indicates that if the object is a pan in the upside down pose then no task should be executed, as grasping the object in this situation is very difficult.

If M is the CP-theory for task-dependent grasping, we can *query for the most likely semantic pre-grasp* of an object. This is equivalent to calculating $argmax_{Pt} P(grasp(o, t_2, Pt)|M)$, where Pt is a part in the set of observed object parts and t_2 is the given task. For the example in Figure 6.3 the distribution over possible parts when the task considered is *pass* becomes: $P(grasp(o, pass, middle)) = 0.87$, $P(grasp(o, pass, top)) = 0.08$, $P(grasp(o, pass, bottom)) = 0.03$, $P(grasp(o, pass, handle)) = 0.01$.

Similar to the object categorization theory, there are different levels of generalization with respect to the rules. To test the brittleness of the theory we experimented also with more general rules, by generalizing over the object pose and containment with respect to several tasks and thus, reducing the number of rules. For example, we replaced part of the theory presented in Example 26 for task *pass* and super-category *dish* with a more general theory illustrated in the following example:

***Example* 28.**
$0.1 :: grasp(X, T, bottom) \leftarrow affords(X, T), pass(T), dish(X), part(bottom, X).$
$0.6 :: grasp(X, T, middle) \leftarrow affords(X, T), pass(T), dish(X), part(middle, X).$
$0.2 :: grasp(X, T, top) \leftarrow affords(X, T), pass(T), dish(X), part(top, X).$
$0.1 :: grasp(X, T, handle) \leftarrow affords(X, T), pass(T), dish(X), part(handle, X).$

We have defined our models using human experience and "educated guesses". They can be augmented by adding extra rules to include new object/task categories. The world knowledge was encoded as general as possible while still reflecting the ontologies and task-object affordances. The parameters of the rules composing the models can, in principle, be learned from data [Meert et al., 2008] to best represent the application domain. Our current experimental results with the quite rigid affordance model can be improved by learning better probability estimates for object-task affordances from data.

6.2.7 Shape-based Grasping

The probabilistic logic module selects the pre-grasp and/or the task to be performed. In order to execute the grasping, the third module of the pipeline (cf. third module box in Figure 6.4) employs additional shape features characterizing

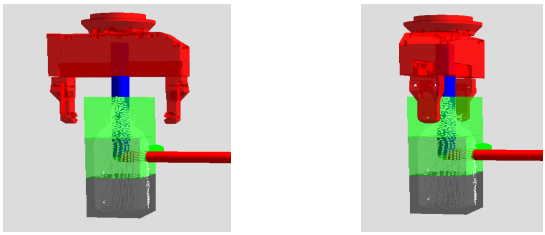


Figure 6.9: Examples of the pre-grasp gripper poses for a face of the top part of a bottle.

the object part to select optimal grasping poses. There are several possible ways to consider shape features at this stage. We compute the probabilities of the grasping pose hypotheses defined by the bounding box of the object part. One bounding box generates several hypotheses, providing two grasping poses for each face of a box, as illustrated in Figure 6.9. The shape-based module calculates $P(grasp|local\ shape)$ by mapping the classification output of an SVM onto a probability. The SVM classifier discriminates between graspable and non-graspable shapes. The pose with the highest grasping probability is selected. Further, the grasping approach directions are defined by the normals to the face of the bounding box that defines the best grasping pose. Since each face has two possible poses, we consider both, however, one will be discarded by the collision checker and the motion planner. Thus, the final number of grasping hypotheses is pruned in a first stage by the task-dependent probabilistic logic module and the shape-based module, and in a second stage by a collision checker and the motion trajectory planner.

Depth difference features

The local shape features are computed in the volume enclosed by the gripper, which is a bounding box located and oriented according to the pre-grasp hypothesis pose. Depth changes in the objects were shown helpful to recognize graspable regions, even in cluttered environments where objects cannot be segmented accurately [Fischinger et al., 2013]. We employ a feature with computations based also on heights, yet, it can be computed for any grasping orientation. This scales better to the diversity of object parts we consider in our experiments than the symmetry height accumulated feature [Fischinger et al., 2013] which is robust, but constrained to top grasps only. Our feature, called *depth gradient image* (DGI), computes the gradient of the depth image in the volume enclosed by the gripper. This volume defines a depth value, that

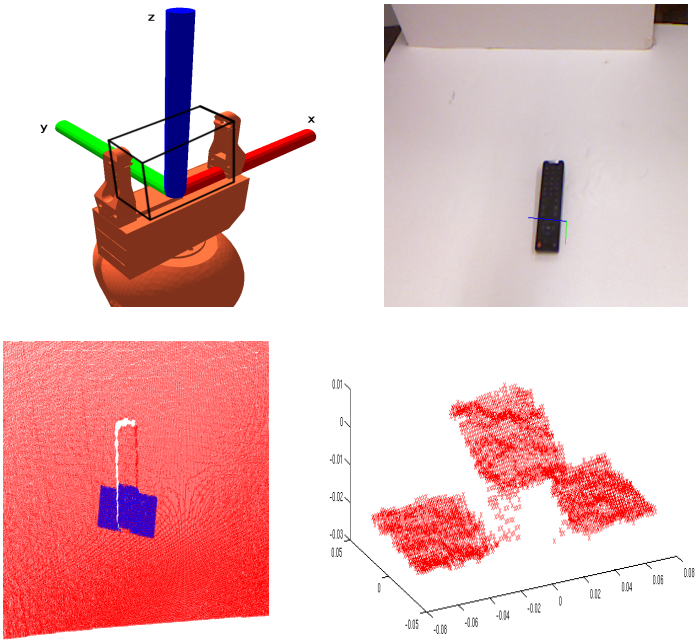


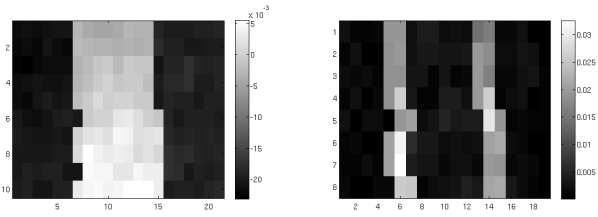
Figure 6.10: Gripper and volume of interest, showing the reference frame origin for the orthogonal projection of the DI image from Eq. (6.3) (top left). Object (top right) and its correspondent point cloud (bottom left). The blue points show the selected points of a graspable region of the remote control. The bottom right image shows the points enclosed by the gripper volume.

is, the Z-component of the distance from the gripper base to the object point. Figure 6.10 shows an example of the selected region of an object and the volume of interest enclosed by the gripper. The DGI acts as a local shape descriptor for grasping prediction. The descriptors of several graspable and non-graspable regions are next fed into the SVM classifier to obtain a shape-based grasping model for grasping hypothesis selection.

The depth image (DI) requires a discrete sampling of the volume, which is described as boxes of $7 \times 7 \times 15$ (mm). A depth image sample is defined as:

$$\mathrm{DI}(u,v) = \begin{cases} \min\{z\} & \text{if } z \in \mathrm{box}(u,v) \\ -1 & \text{otherwise,} \end{cases} \tag{6.3}$$

where $\mathrm{box}(u,v)$ represents the set of points inside the box defined by the pixel (u,v). Eq. (6.3) performs an orthogonal projection of the closest point to the base of the gripper for every pixel in the depth image. Figure 6.11 shows the



(a) Depth Image (b) Depth Gradient Image

Figure 6.11: Example of a depth image (10x21 pixels) and its corresponding gradient magnitude (8×19 pixels).

depth image for the selected volume in Figure 6.10. The DGI is computed on the depth image by applying pixel differences in u and v as follows:

$$\mathrm{DI}_u(u,v) = \mathrm{DI}(u+1,v) - \mathrm{DI}(u-1,v), \tag{6.4}$$

$$\mathrm{DI}_v(u,v) = \mathrm{DI}(u,v+1) - \mathrm{DI}(u,v-1), \tag{6.5}$$

$$\mathrm{DGI}(u,v) = \sqrt{\mathrm{DI}_u(u,v)^2 + \mathrm{DI}_v(u,v)^2}. \tag{6.6}$$

Grasping probability

Given DGI shape features x_i and their labels y_i, we use SVMs [Cortes and Vapnik, 1995] with the RBF kernel to discriminate between successful and a failed grasps. Before applying the sign function, we map the SVM output onto a probability by applying a sigmoid function to the decision value from Eq. (6.7).

$$h(x_i) = \langle w, \phi(x_i) + b \rangle, \tag{6.7}$$

where (6.8)

$$\phi(x_i) = \exp(-\gamma |x_i - x_j|^2), \tag{6.9}$$

We employ the parametric sigmoid

$$P(grasp|local\ shape) = \frac{1}{1 + \exp(A \cdot h(x_i) + b)}\ , \tag{6.10}$$

where the parameters A and b are obtained by generating a hold-out set and cross-validation. Its advantages were shown empirically in [Platt, 1999]. We

trained the SVM classifier on the grasping rectangle dataset [Jiang et al., 2011], available at `http://pr.cs.cornell.edu/placingobjects/`. Our approach using only depth features has good performance, improving the result reported in [Jiang et al., 2011].

6.3 Experiments

We address experimentally the benefits of our probabilistic logic module and the performance of the full pipeline for robot grasping. Specifically, we investigate the following main questions:

(Q1) How robust is the probabilistic logic module w.r.t. the considered robotic tasks? How well does it cope with missing information?

(Q2) Does the integration of high-level reasoning (about task and object category) and low-level learning improve the grasping performance upon local shape features?

We decompose (Q1) into the following questions, which we answer in turn:

(i) does the probabilistic logic module improve upon manifold information for object category prediction?

(ii) can it predict correctly suitable tasks?

(iii) can it predict the correct pre-grasp region?

We investigate its robustness with respect to object category, task, pre-grasp and pre-grasp pose in Subsection 7.4.3. We perturb either the visual observations about the world by dropping the prior on the object category, or the CP-theory by keeping the more general rules.

For question (Q2) we first learn a classifier that maps pre-grasp poses to successful grasps using local shape features. Given a new object, we then directly predict the most likely pre-grasp pose using solely local shape information. This is our local shape-based baseline. We compare the baseline with the pipeline classifier, which maps pre-grasp poses extracted from predicted semantic pre-grasp regions (or object parts) to successful grasps, using similar local shape features. Given a new object, we first predict, using high-level reasoning, the most likely object pre-grasps. We then use the classifier to predict good grasping poses among the set of possible grasping poses in the inferred pre-grasp.

6.3.1 Datasets and evaluation scenarios

We consider three types of datasets to quantitatively investigate the robustness and power of generalization of our SRL approaches. For the first type, the object point clouds are obtained from 3D meshes and the object parts are manually labeled. In this case the dataset is synthetic and actual grasps are not executed. For the second type, data samples are obtained from the ORCA simulator [Baltzakis, 2005], while for the third type, data samples are obtained from a real robot platform.

Synthetic scenario

We first consider flawless object point clouds obtained from 3D meshes. The object points are distributed uniformly on the object surface according to their size by applying the midpoint surface subdivision technique. Point normals are correctly oriented, the object pose and its parts are manually labeled as well as the object containment. This "perfect scenario" serves as an upper-bound comparison scenario to the more realistic scenarios, allowing an extensive evaluation of the generalization capabilities of the probabilistic logic module. The dataset contains 41 objects belonging to all categories in our ontology and 102 grasping scenarios. We denote this dataset S_{SYN} and we use it to evaluate the probabilistic logic module.

ORCA scenarios

The second type of datasets is used to evaluate the full pipeline, the pipeline modules and the grasping point prediction in simulation. We use ORCA, which provides the sensors (laser range camera Asus Xtion PRO and the Universal Gripper WSG 50 force sensor), robotic arm (KUKA LightWeight Robot), objects and interface to physics engine (Newton Game Dynamics library) for robot grasping simulation. The other modules are external to ORCA and interfaced both with the simulated and real robot. These modules include: object completion (where needed), part and pose detection, global shape similarity, probabilistic logical reasoning modules, local shape feature construction and grasping prediction and the tree-based motion planner [Sánchez and Latombe, 2003] available in the Open Motion Planning Library (OMPL) [Şucan et al., 2012]. Each object is placed on top of a table.

Our simulated robot scenario (see Figure 6.1) considers four possible settings for the grasping pipeline. In the first setting, object pose is not estimated but given by the ground truth, while the parts are estimated from the completed

point cloud, as explained in Section 6.2.2. Thus, the scene description may have missing parts when they are occluded or not detected. Additionally, we assign to all detected parts probability 1.0. We denote this dataset S_{REAL_semi}. In the second setting, both object pose and its parts are estimated from the completed point cloud. While the pose has associated a likelihood, we keep highly confident parts. We denote this setting S_{REAL}. In the third setting we also provide a part likelihood according to the limitations of the detection algorithm. We denote this dataset S_{REAL_noisy}. Finally, the fourth setting includes actual grasping tests with the simulated robot. It comprises a subset of the scenarios from the third setting, where all containers are empty and all objects are graspable by the robot. The rationale behind is that it is very difficult to check whether a container is full or if some objects (too big or too small) do not fit the gripper capabilities. We denote this dataset S_{GRASP_noisy}. In addition, object poses considered are *upright* or *sideways* due to the ambiguity between upright and upside-down when using global shape representations. Each of the first three settings contains 26 different objects, instances of categories *pan, bowl, cup, glass, bottle, can, hammer, screwdriver, knife* and 126 grasping scenarios. The fourth setting contains 18 objects, instances of categories *pan, cup, glass, bottle, can, hammer, screwdriver, knife* and 113 grasping scenarios.

Real robot scenario

The real robot scenario (see Figure 6.1) considers the same type of tests as those included in the S_{GRASP_noisy} and is used for the evaluation of our task-dependent grasping pipeline. In addition, we evaluate the performance of the pipeline when two or more objects are in the field of view of the camera and the field of action of the arm, considering three settings of increasing complexity in terms of path planning. The less complex setting (scenario1), considers only two objects which are instances of *glass* and *bottle*, in a way that planning constraints are very similar to a single object on the table. The setting with intermediate complexity (scenario2) includes three objects which are instances of *can, hammer* and *screwdriver*. The more complex setting (scenario3) considers four objects which are instances of *bottle, glass* and *cup*. Figure 6.12 shows the objects of every scenario. In addition to the larger number of objects, we also consider object placement as another criterion for evaluation. Object placement is performed in two steps: (i) plan from the grasp pose to a post-grasp pose and (ii) plan from the post-grasp pose to the grasp pose. We denote the associated dataset S_{ROBOT}.

The synthetic dataset and part of ORCA datasets are available for download at `http://www.first-mm.eu/data.html`.

Figure 6.12: Experimental settings with the real robot. Each picture shows the objects utilized for each scenario. Additional object constraints are: the gray bottle of scenario3 is full with water, the white bottle is empty and the coffee container is full of coffee.

6.3.2 Evaluation measures

We evaluate our experiments in terms of accuracy given by $\frac{\#successes}{\#tries} \cdot 100\%$. We assess a *success* in several ways. Depending on the prediction task, the ground truth is either one value (object categorization) or a set of values (task and pre-grasp prediction, part detection). For object categorization we take as prediction the category with the highest probability and consider it a success if it matches the ground truth category. For the uniform prior, it can be that two or more categories are predicted with the same probability. This case is reported as a false positive.

For task/pre-grasp prediction evaluation, the ground truth Gt of each instance is a set (e.g., the tasks *p&pOn* and *p&pInUpright* may be equally possible in a particular scenario). In this case, we compare the set of best predictions Pr to Gt, where $|Pr| \leq |Gt|$. If $Pr \subseteq Gt$ a success is reported. We present results for different sizes of Pr, such that $|Pr|$ belongs to the set $\{|Gt| - i\}$, with i ranging from 0 (the most restrictive evaluation setting) to $|Gt| - 1$ (the most pertinent setting). We denote the possible evaluation settings as E_i. For the scenarios with robot grasping execution (S_{GRASP_noisy} and S_{ROBOT}) the evaluation must consider the success of grasping execution with respect to the valid grasping hypotheses provided by the pipeline. In this case the accuracy is given by $\frac{\#correctly\ estimated\ task/part}{\#valid\ grasping\ hypotheses} \cdot 100\%$.

6.3.3 Results and discussion

In the following we present quantitative experimental results for all questions. For all results bold font, if present, indicates the best performance.

(Q1) We investigate the robustness of the probabilistic logic module w.r.t. the considered robotic tasks by evaluating the performance of each of its components.

(Q1-i) *Object category prediction.* We first evaluate the object category prediction task and report accuracy results in Table 6.2. We compare random category assignment (Random), propagation kernels (Manifolds) and the probabilistic logic module (PLM). For Manifolds we set the parameter T to give the best leave-one-out accuracy performance.[‡] We observe that, overall, manifold information leads to a good prior distribution among object categories (more details can be found in [Neumann et al., 2013]).

Next, we incorporated the manifold priors into the PLM. We started from the priors that gave the best accuracy for each task. To show the robustness of the PLM, we varied the generality of our categorization theory and experimented with and without the manifold prior on the object category. $PLM^{general}_{uniform}$ indicates the more general theory setting with the uniform prior, while $PLM^{general}_{manifold}$ indicates the more general theory with the manifold prior. The results show that the PLM improves object categorization accuracy upon manifold information. By increasing the generality of the theory, we do not loose much in terms of performance when the manifold information is used, and we are still able to improve upon the prior.

Note that by removing the manifold prior, the $\text{PLM}_{manifold}$ still gives a reasonable result (3 times better than Random on S_{SYN}). We also evaluated the $\text{PLM}_{uniform}$ with a second accuracy definition (acc) in which a *success* is reported if at least one category in the set of equally and maximum category prediction values is equal to the ground truth. In this case we obtain $acc = 68.25\%$ (S_{REAL} datasets) and $acc = 99.02\%$ (S_{SYN}) for $\text{PLM}_{uniform}$, and $acc = 62.70\%$ (S_{REAL} datasets) and $acc = 94.12\%$ (S_{SYN}) for $PLM^{general}_{uniform}$. The results obtained using this evaluation setting explain the good performance for the other grasping tasks explained in the following. That is, estimating the category of an object as any of the sub-categories of a super-category in the ontology is satisfactory to predict good semantic pre-grasps.

(Q1-ii) *Task prediction.* We investigate *what is the suitable task* by reporting results in Table 6.3 (top rows) with evaluation setting E_0 for both general and more specific object categorization theories. Figure 6.13 (left) presents results for the specific object categorization theory using all evaluation settings. We experimented on all datasets with and without the manifold prior. The presence of the prior gives significantly better results in the most restrictive evaluation setting. For the other evaluation settings, in most situations, the probabilistic

[‡]The Manifolds results for S_{REAL_semi}, S_{REAL} and S_{REAL_noisy} are the same due to the fact that object pose and part confidence are not used by the propagation kernel. We refer to these datasets as S_{REAL} for the Manifolds evaluation.

Dataset	Random	Manifolds	PLM$_{uniform}$	PLM$^{general}_{uniform}$	PLM$_{manifold}$	PLM$^{general}_{manifold}$
S_{SYN}	9.1	87.8	31.37	31.37	**93.14**	92.16
S_{REAL_semi}	9.1	39.7	14.29	14.29	**49.21**	*46.83*
S_{REAL}	9.1	39.7	14.29	14.29	**48.41**	*46.83*
S_{REAL_noisy}	9.1	39.7	14.29	14.29	**39.7**	**39.7**

Table 6.2: Accuracy (%): PLM vs. propagation kernel (Manifolds) vs. random baseline for object categorization.

Dataset	$\text{PLM}_{uniform}$	$\text{PLM}^{general}_{uniform}$	$\text{PLM}_{manifold}$	$\text{PLM}^{general}_{manifold}$
S_{SYN}	71.57	65.69	**72.55**	**72.55**
S_{REAL_semi}	**98.41**	**98.41**	95.24	93.65
S_{REAL}	80.16	80.16	**95.24**	93.65
S_{REAL_noisy}	38.10	38.10	**93.65**	**93.65**
S_{GRASP_noisy} E_0	30.00	-	**35.71**	-
S_{GRASP_noisy} E_1	40.00	-	**50.00**	-
S_{ROBOT} E_0	**28.57**	-	25.00	-
S_{ROBOT} E_1	**85.71**	-	75.00	-

Table 6.3: Accuracy (%): PLM for task prediction.

logic module will return, although not as the first option, a correct possible task with or without a prior.

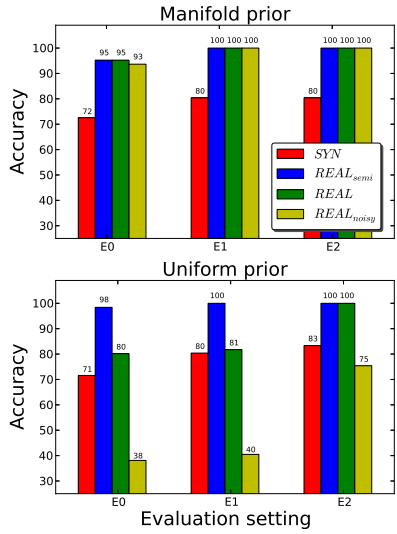
In the scenarios with grasp execution (bottom rows in Table 6.3), the evaluation settings E_0 and E_1 consider the outcome of the grasping action. The additional source of failures on grasping include uncertainty on the pose of the objects and the gripper, which are caused by the sensor and the object completion. These sources have effects on the performance of the planner, for instance placing the gripper a bit misaligned or hitting and object before closing the gripper. The few cases where the uniform prior provided better results (S_{ROBOT}) are explained by the fact that the complete pipeline failed on the $\text{PLM}_{manifold}$ experiment more than the uniform prior due to uncertainty.

We note the importance of having a prior probability distribution over the object categories, rather than the top category. We perform the same experiments only with the top predicted category and obtain accuracies of 95.24%, 89.68% and 89.68% for S_{REAL_semi}, S_{REAL} and S_{REAL_noisy}, respectively, which are lower than using the full prior (see Table 6.3).

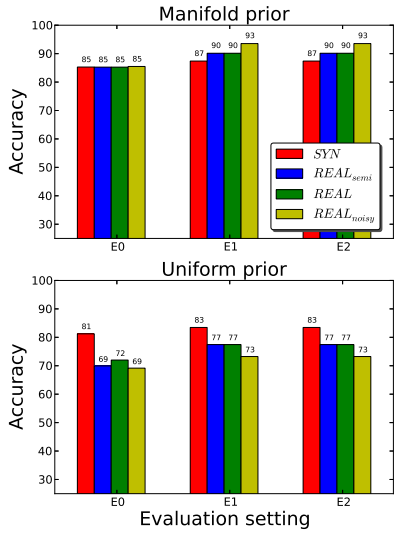
(Q1-iii) *Pre-grasp prediction.* For this question we considered the specific object categorization theory and experimented with both a more specific and a more general task-dependent grasping theory. The results for both settings when the task is given for evaluation setting E_0 are shown in Table 6.4. We note that increasing the generality of the model does not cause much performance loss. The result confirms generalization over similar object parts and object/task categories, which implies that if the input object is an unseen category, such as a paint roller or a vase the grasping pipeline is robust enough to return a good grasping part. This allows us to experiment with a wide range of object/task categories and lets us to believe that our approach can be extended beyond the categories used, by augmenting the probabilistic logic module with extra rules.

Dataset	$\text{PLM}_{uniform}$	$\text{PLM}^{general}_{uniform}$	$\text{PLM}_{manifold}$	$\text{PLM}^{general}_{manifold}$
S_{SYN}	81.23	80.67	**85.29**	84.73
S_{REAL_semi}	69.95	72.00	85.26	**86.73**
S_{REAL}	72.00	72.00	**85.26**	84.69
S_{REAL_noisy}	69.16	69.16	85.49	**86.73**
S_{GRASP_noisy} E_0	66.7	-	**75.51**	-
S_{GRASP_noisy} E_1	66.7	-	**75.51**	-
S_{ROBOT} E_0	**66.7**	-	**66.7**	-
S_{ROBOT} E_1	**66.7**	-	**66.7**	-

Table 6.4: Accuracy (%): PLM for pre-grasp prediction.



(a) PLM for task prediction.



(b) PLM for pre-grasp prediction.

Figure 6.13: Accuracy (%) of PLM for task and pre-grasp prediction using all evaluation settings.

When the task is predicted by the probabilistic logic module we report accuracies of 95.10% and 100.0% for S_{SYN} and S_{REAL_noisy}, respectively, for both priors using the evaluation setting E_0. When the task is given, results for the pre-grasp prediction task using the most specific task-dependent grasping theory for all evaluation settings are reported in Figure 6.13 (right).

For this task as well, in the scenarios with grasp execution, the evaluation considers the outcome of the grasping action. In all cases the object category provided by the manifold matching has better performance than the uniform category prior, showing the benefits of the pipeline. Again, it is important to consider the full prior distribution as input to the PLM, instead of only the top category. Our experiments using solely the top category resulted in accuracies of 81.63, 81.63 and 82.88, respectively, for the three S_{REAL} datasets, which are lower than the ones using the full prior.

The robot experiments (S_{GRASP_noisy} and S_{ROBOT}) did not consider the mutually exclusiveness assumption. However, for the other datasets we performed experiments both with and without this assumption. The means of the differences between the mutually exclusive and the non-exclusive results for object categorization, task prediction and pre-grasp prediction (1.6%, 8.5% and 2.6%, respectively) indicate a better result for the mutually exclusive setup. This lets us to believe that current robot results can be only improved by the exclusiveness assumption.

(Q2) We acknowledge whether the integration of high-level reasoning and low-level learning improves grasping performance upon local shape features by evaluating the full pipeline.

Full pipeline evaluation. The selection of grasping points using only local shape descriptors bias the ranking of the object points towards the most "visually graspable", disregarding other constraints such as the pre-grasp pose for task execution, path planning and post-grasp object pose. By adding those constraints, regions with lower "visually graspable‘" probability will become more important when considering the task execution probability and vice versa. Thus, we compute the percentage of grasping points that have a low "visually graspable" probability as calculated in Section 6.2.7, but still lead to successful grasps when taking into account task constraints. Results are shown in Table 6.5 (Pipeline$_{manifold}$), having as baseline the local shape-based approach. We also investigate its robustness with respect to grasping point prediction by dropping the prior on the object category (Pipeline$_{uniform}$).

The full pipeline selects in average more points having low "visually graspable" probabilities than the other options. This behavior confirms the importance of task constraints on the computation of the grasping probability. We note that the complete pipeline clearly improves upon using the local shape-based approach. This answers affirmatively (Q2). Additionally, the results obtained by Pipeline$_{uniform}$ show again the robustness of the pipeline.

Finally, we present results for the grasp and place action for each level of

Feature (DGI)	Measure	Local	Pipeline$_{manifold}$	Pipeline$_{uniform}$
S_{GRASP_noisy}	lt 0.5	44.55%	**52.38%**	50%
	lt 0.4	27.27%	**28.57%**	21.15%
	lt 0.3	4.55%	**7.94%**	7.69%
S_{ROBOT}	lt 0.5	37.5%	**50%**	46.15%
	lt 0.4	25%	**42.86%**	23.08%
	lt 0.3	12.5%	**28.57%**	23.08%

Table 6.5: Percentage (%) of successfully graspable points that have "visually graspable" probability less than (lt) 0.3, 0.4 or 0.5: Pipeline vs. local shape grasp prediction.

Scenario	Total tests pre-grasps	Reachable objects	Grasped objects	Placed objects
scenario1	10	9 (90%)	7/9 (77.8%)	7/9 (77.8%)
scenario2	15	10 (66.7%)	7/10 (70%)	7/10 (70%)
scenario3	20	16 (80%)	8/16 (50%)	8/16 (50%)

Table 6.6: Percentage of successful grasps in the real robot scenarios. Different levels of S_{ROBOT} complexity.

complexity on the S_{ROBOT} dataset in Table 6.6. It is important to notice the performance drop of 20% in average when the complexity increases from medium (scenario2) to complex (scenario3). Considering that the multiple object scenario of these experiments is rather simple, one way to improve the performance would be to plan for object displacement (sequence of actions before the pre-grasp pose) before the grasp execution in order to increase the grasping performance on scenario3.

We note that in all our experiments with the pipeline *the result changes if the data is perturbed* for the task, pre-grasp and the grasping point prediction. This highlights the importance of the object categorical information for robot grasping. However, when this information is absent the probabilistic logic theory is robust enough to give reasonable results thanks to the use of world general knowledge. The experiments show the *robustness of the probabilistic logic module when more general rules* are used.

6.4 Related work

Object grasping is an important problem in robotics. We review the related work following three main directions. First, we address visual-dependent grasping. Next, we present the related work on task-dependent grasping. Finally, we give an overview of the existing work that employs SRL techniques in robotics.

6.4.1 Visual-dependent grasping

A significant number of vision-based grasping methods learn a mapping from 2D/3D features to grasping parameters [Saxena et al., 2008b, Bohg and Kragic, 2010, Erkan et al., 2010, Kraft et al., 2010, Jiang et al., 2011, Montesano and Lopes, 2012, Lenz et al., 2013]. However, it is difficult to link a 3D gripper orientation to local image features without considering contextual or global object information. Only recently, methods that take more global and symbolic information into account have been proposed [Aleotti and Caselli, 2011,Neumann et al., 2012a]. They benefit from increased geometric robustness, which gives advantages with respect to the pre-shape of the robotic hand and general shape of the object, generating more accurate grasps. Nevertheless, global information is exclusively dependent on a complete shape of the object.

Object reconstruction based on single views is a difficult problem due to lack of observability of the self-occluded parts and requires several object-related assumptions. We combine task-dependent grasping with a shape completion method based on the symmetry assumption inspired by the work of Thrun and Wegbreit [Thrun and Wegbreit, 2005] and similar in spirit with Bohg et al. [Bohg et al., 2011]. The new computationally efficient shape completion approach we incorporated into the proposed probabilistic logic pipeline translates a set of environmental assumptions into a set of approximations, allowing us to reconstruct the object point cloud in real-time, given a partial view of the object.

6.4.2 Task-dependent grasping

Since grasping is highly correlated with the task to be performed on the object, recent work has focused on incorporating task constraints in robot grasping. This is mostly done by learning a direct mapping function between good grasps and geometrical and action constraints, action features and object attributes. A part of this work focuses on Bayesian network learning to integrate symbolic task goals and low-level continuous features such as object attributes, action

properties and constraint features [Madry et al., 2012a, Song et al., 2010]. The goal is to learn features of importance for grasping knowledge transfer. This work is extended to consider object categorical information as an additional feature to predict suitable task-dependent grasping constraints [Madry et al., 2012b]. Further, Detry et al. [Detry et al., 2013, Detry et al., 2012b, Detry et al., 2012a] identify grasp-predicting prototypical parts by which objects are usually grasped in similar ways. The discrete part-based representation allows robust grasping. Differently, in addition to the semantic parts, we also consider a task-dependent setting that uses probabilistic logic and world-knowledge to reason about best pre-grasps. Several approaches make use of object affordances for grasping. While in [Sweeney and Grupen, 2007] the authors employ estimated visual-based latent affordances, the work in [Barck-Holst et al., 2009] reasons about grasp selection by modeling affordance relations between objects, actions and effects using either a fully probabilistic setting or a rule-based ontology. In contrast, we employ a probabilistic logic-based approach to task-dependent grasping which can generalize over similar object parts and several object categories and tasks.

Related to our probabilistic logic pipeline is the fully probabilistic one introduced in [Bohg et al., 2012]. It combines low-level features and Bayesian networks to obtain possible task-dependent grasps. Closely related is the semantical pipeline presented in [Dang and Allen, 2012]. It employs a semantic affordance map which relates gripper approach directions to particular tasks. However, we exploit additional object/task ontologies using probabilistic reasoning and leverage low-level learning and semantic reasoning. This allows us to experiment with a wide range of object categories.

6.4.3 SRL for robot grasping and other robotic tasks

From a different point of view, probabilistic relational robotics is an emerging area within robotics. Building on statistical relational learning (SRL) and probabilistic robotics, it aims at endowing robots with a new level of robustness in real-world situations. We review some recent successful contributions of SRL to various robotic tasks. Probabilistic relational models have been used to integrate common-sense knowledge about the structure of the world to successfully accomplish search tasks in an efficient and reliable goal-directed manner [Hanheide et al., 2011]. Further, relational dependency networks have been exploited to learn statistical models of procedural task knowledge, using declarative structure capturing abstract knowledge about the task [Hart et al., 2005]. The benefits of task abstraction were shown in [Winkler et al., 2012], where the robot uses vague descriptions of objects, locations, and actions in combination with the belief state of a knowledge base for reasoning. The goal of this work is to robustly solve the planning task in a generalized pick and place

scenario. Abstract knowledge representation and symbolic knowledge processing for formulating control decisions as inference tasks have proven powerful in autonomous robot control [Tenorth and Beetz, 2009]. These decisions are sent as queries to a knowledge base. SRL techniques using Markov Logic Networks and Bayesian Logic Networks for object categorization from 3D data have been proposed in [Marton et al., 2009].

In probabilistic planning, relational rules have been exploited for efficient and flexible decision-theoretic planning [Lang and Toussaint, 2010] and probabilistic inference has proven successful for integrating motor control, planning, grasping and high-level reasoning [Toussaint et al., 2010]. In mobile robotics, relational navigation policies have been learned from example paths with relational Markov decision Processes [Cocora et al., 2006]. In order to compute plans comprising sequences of actions and in turn be able to solve complex manipulation tasks, reasoning about actions on a symbolic level is incorporated into robot learning from demonstrations [Abdo et al., 2012]. Symbolic reasoning enables the robot to solve tasks that are more complex than the individual, demonstrated actions. In [Kulick et al., 2013] meaningful symbolic relational representations are used to solve sequential manipulation tasks in a goal-directed manner via active relational reinforcement learning. Relational Markov networks have been extended to build relational object maps for mobile robots in order to enable reasoning about hierarchies of objects and spatial relationships amongst them [Limketkai et al., 2005]. Related work for generalizing over doors and handles using SRL has been proposed in [Moldovan et al., 2013a]. All of these approaches successfully intertwine relational reasoning and learning in robotics. However, none of these frameworks solves the generalization capability needed for task-dependent grasping following an affordance-based behavior. Relational affordance models for robots have been learned in a multi-object manipulation task context [Moldovan et al., 2012]. We propose a probabilistic logic pipeline to infer pre-grasp configurations using object-task affordances.

6.5 Conclusions and Future Work

This chapter makes two contributions to robot grasping. The first is a new probabilistic logic pipeline which combines high-level reasoning and low-level learning for task-dependent grasping. The high-level reasoning leverages symbolic world knowledge, in the form of object/task ontologies and object-task affordances, object categorical and task-based information. The low-level part is based on learning with shape features. The non-trivial realization of high-level knowledge relies on logic, which exploits world knowledge and relations to encode compact grasping models that generalize over similar object parts and

object/task categories in a natural way. When combined with probabilistic reasoning, our proposed pipeline shows robustness to the uncertainty in the world and missing information. In addition, our experiments confirm the importance of high-level reasoning and world-knowledge as opposed to using solely local shape information for robot grasping.

6.5.1 Future Work

As future work we point out several important directions. One of them is learning the parameters and structure of our grasping CP-theory from data. Further, we would like to extend our object/task ontologies and expand the grasping modules to be able to generalize across more object categories and tasks. Finally, another direction is planning the sequence of actions in order to fulfil the task-dependent pre-grasp poses. Since planning in presence of multiple objects raises complexity and generalization issues, considering relational planners similar to those in [Moldovan et al., 2013b] as part of the probabilistic logic module may provide successful plans for pre-grasp tasks.

Chapter 7

Relational Kernel-based Grasping with Numerical Features

As the previous chapter shows, our experiments with the proposed probabilistic logic pipeline confirm that performing a grasp depends on high-level world-knowledge, the object (e.g., its shape), and grasp constraints (e.g., gripper configuration, environmental restrictions). To execute the grasping, the third module of the pipeline (cf. third module box in Figure 6.4) employed additional shape features characterizing the object part to select optimal grasping poses. While there we selected the best grasping pose hypotheses based on the bounding boxes of the relevant semantic object parts, providing a statistical and purely appearance-based solution, in this chapter we present an new alternate SRL solution to recognize optimal grasping poses.

Our contribution builds on the reaching point concept [Erkan et al., 2010,Popovic et al., 2010]. In this setup, every point of the object point cloud defines the position of the end gripper effector and its approaching orientation. The reaching point concept places the gripper at a constant distance from the point in the direction of the surface normal at that point. This reduces the number of hypotheses for grasping execution in two ways. First, the gripper parameters are given by the local geometry of the point cloud. Second, the collision checking between the gripper and the point cloud reduces largely the number of points selected for grasping execution. The points where the gripper does not collide with the point cloud provide samples for learning to recognize good grasping

points in object clouds. Thus, robot grasping heavily relies on finding good mappings between gripper orientations and object points. However, it is a challenging problem to learn fully unstructured models that directly map local visual perceptions to good grasps. This may be the case if the robot acts in highly dynamical real-world environments handling objects that belong to a large range of different categories, such as household or supermarket objects.

Current work mainly focuses on adapting low-level descriptors popular in the computer vision community (i.e., shape context) to point cloud representations for grasping. For each point in the cloud, an appearance-based feature descriptor characterizing a limited neighboring surface around the point is calculated. Although they give acceptable results, such features are restricted to a local description of the point and do not capture enough contextual information about the object. Instead, as one contribution, we investigate whether *the structure of the object can improve robot grasping using SRL.* As an example, consider a graspable point on the rim of a cup. Although it may be characterized by a misleading local shape descriptor due to its position or perceptual noise, this can be corrected by nearby graspable points with more accurate shape features.

As our second contribution, we detect graspable points by *combining, in a kernel-based manner, numerical appearance features with qualitative spatial relations among them.* Given a (partial) 3D point cloud, we characterize each point with shape features and represent each cloud as a (hyper-) graph by considering symbolic spatial relations between neighboring points. Further, we use kernels on graphs (i.e., kLog) to exploit extended contextual shape information and compute highly discriminative features which show improvement upon local shape features. Our work for robot grasping highlights the importance of moving towards integrating relational representations with low-level descriptors for robot vision. The result is a *new relational kernel-based approach to numerical feature pooling for robot grasping.* Its benefit is shown experimentally on a realistic dataset. Our SRL approach outperforms the performance of solely shape features on the same task. A related kernel-based approach for graphs was successfully employed in Chapter 6 for object categorization. There, the kernel function was defined over counts of similar continuous node label distributions and relied on semantic labels and local sensitive hashing. Differently, this chapter considers shape-based continuous features organized in graph structures. In this case the nodes are characterized by distributions of appearance features instead of distributions of semantic labels. The propagation kernel has not yet been employed for such appearance-based continuous input and, due to its nature, it does not guarantee successful results.

We proceed as follows. We first explain in Section 7.1 the grasping primitives that define our setup. Afterwards, we present our relational formulation for the learning problem considered (Section 7.2) and show how we solve it with

variants of relational kernels (Section 7.3). Next, in Section 7.4 we present our experimental results. Before concluding, we review related work on robot grasping, feature pooling and graph kernels 7.5.

7.1 Robot grasping primitives

We consider three types of domain primitives which we use to build our relational representation (or hyper-graphs) of the grasping problem: *reaching points*, their *3D locations* and their *shape features*. Reaching points are labeled as good grasping points using the simulator. The robot executes grasps on the object points and if they are successful, the reaching points become positive instances. Next, each reaching point is characterized by several local 3D shape features computed in its neighborhood. The neighborhood of each point consists of a 3D grid centred at the reaching point and oriented with respect to the projection of the point's normal on the table plane and the gravity vector, as illustrated in Figure 6.2. We consider as neighborhood grid, in turn, the gripper cell and a sphere around the point and calculate three shape features: 3D shape context (SC) [Körtgen et al., 2003], point feature histogram (PFH) [Rusu, 2009] and viewpoint feature histogram (VFH) [Rusu et al., 2010].

While the PFH feature encodes the statistics of the shape of a point cloud by accumulating the geometric relations between all point pairs, the VFH augments PFH with the relation between the camera's point of view and the point cloud of an object. The 3D SC describes the structure of the shape as relations between a point to the rest of the points in the region. Given the coordinates of a point p on the shape, the shape context descriptor is constructed as a histogram of the direction vectors from p to the rest of the points.

7.2 Relational Grasping Problem Formulation

Next, we use the grasping primitives as input to our relational learning system. We use the kLog framework [Frasconi et al., 2012] introduced in Chapter 4 to employ a relational kernel-based approach to grasping point recognition. As already explained, kLog is a domain specific language for kernel-based learning, embedded in Prolog that allows to specify in a declarative way relational learning problems at a high level using E/R models. Its declarative nature allows us to construct and integrate multiple heterogeneous features and specify relational learning problems. It transforms the created relational databases into graph-based representations and uses graph kernels to extract the feature space. The

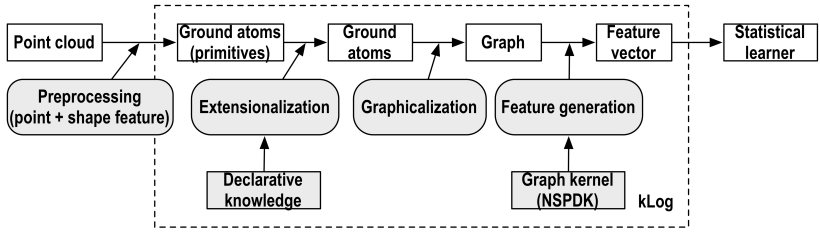


Figure 7.1: From point clouds to feature vectors in kLog.

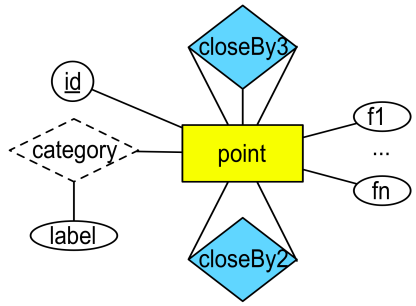
actual features are determined by the choice of the graph kernel and is explained for our grasping problem in Section 7.3.

Figure 7.1 illustrates the information flow in kLog for robot grasping. We model our graspable point recognition problem starting from the grasping primitives which we represent as relational databases. Next, we define declaratively spatial relations between reaching points. The extended relational database is used by kLog to build kernel features which are finally used for learning. We explain in more detail each step for our grasping problem.

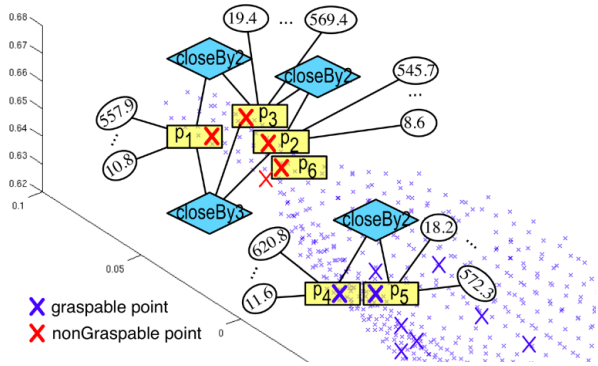
7.2.1 Data modeling

We represent grasping primitives at a higher level using a relational language derived from its associated E/R data model. It is based on entities, relationships linking entities and attributes that describe object points and relationships. Figure 7.2(a) shows the E/R diagram for our application. A *reaching entity* is any reaching point. It is represented by the relation $\texttt{point}(id, f_1, \ldots, f_n)$, which indicates that it has a unique identifier id (underlined oval) and shape properties. The vector $[f_1, \ldots, f_n]$ represents a shape feature characterizing the reaching point. Each f_i is a shape feature vector component and is represented as an entity attribute. For example, the tuple $\texttt{point}(p_1, 10.8, \ldots, 557.9)$ specifies a specific reaching point entity (depicted yellow in Figure 7.2(b)), where p_1 is its identifier and the other arguments are shape feature components.

Relationships are *spatial relations* among entities (blue diamonds) and are derived from their 3D spatial locations. They impose a structure on reaching entities. An example is the relationship $\texttt{closeBy2}(p_1, p_3)$ which indicates that reaching entities p_1 and p_3 are spatially close to each other. Another relationship is introduced by the predicate $\texttt{category}(id, class)$ (white diamond), which is linked to reaching entities and associates a binary class label *grasp* / *nonGrasp* with every entity.



(a) Proposed E/R scheme: rectangles denote entity vertices, diamonds denote relationships, and circles (except point id) denote local properties.



(b) Part of a *glass* grounded E/R scheme mapped on its point cloud.

$x = \{\texttt{point}(p_1, 10.8, \ldots, 557.9), \texttt{point}(p_2, 8.6, \ldots, 545.7), \texttt{point}(p_3, 19.4, \ldots, 569.4),$
$\texttt{point}(p_4, 11.6, \ldots, 620.8), \texttt{point}(p_5, 18.2, \ldots, 572.3), \ldots, \texttt{closeBy2}(p_1, p_3),$
$\texttt{closeBy2}(p_3, p_2), \texttt{closeBy2}(p_4, p_5), \ldots, \texttt{closeBy3}(p_1, p_2, p_3), \ldots\}.$
$y = \{\texttt{category}(p_1, nonGrasp), \texttt{category}(p_2, nonGrasp), \texttt{category}(p_3, nonGrasp),$
$\texttt{category}(p_4, grasp), \texttt{category}(p_5, grasp), \ldots\}.$

(c) Point cloud interpretation $i = (x, y)$ of the same *glass* view.

Figure 7.2: Relational robot grasping in kLog.

7.2.2 Declarative and Relational Feature Construction

We define intensional relations for our graspable point detection problem using logical rules. We define the relation `closeBy2`/2, which holds between two points that belong to the same object view and are close to each other.

***Example* 29.** *The relation* `closeBy2`$/2$ *is defined as follows:*

$$\texttt{closeBy2}(P_1, P_2) \leftarrow \texttt{point}(P_1, F_{11}, \ldots, F_{1n}),\ \texttt{point}(P_2, F_{21}, \ldots, F_{2n}),\ \texttt{belongsToView}(P_1, V),\ \texttt{belongsToView}(P_2, V),\ \texttt{edist}(P_1, P_2, Dist),\ Dist < \sqcup.$$

The relation $\texttt{edist}(A, B)$ is defined in a similar way and represents the normalized Euclidian distance between 2 points in the 3D space. In practice it is projected on all 3 axes and thresholded on each axis. The threshold $\sqcup$ is a constant calculated for every object as a ratio relative to the normalized object dimensions. The condition $\texttt{belongsToView}(P_1, V)$, $\texttt{belongsToView}(P_2, V)$ specifies that P_1 and P_2 belong to the same object view. Another relation is `closeBy3`$/3$ which holds between three points that belong to the same view and are close to each other.

In our defined setting each point cloud is represented as an instance of a relational database (i.e., as a set of relations), and thus, as a *point cloud interpretation*. Object point clouds are assumed to be independent. An example of a point cloud interpretation is given in Figure 7.2(c).

7.2.3 The Relational Problem Definition

We formulate the learning problem at the relational representation level in the following way: given a training set $D = \{(x_1, y_1), \ldots, (x_2, y_2), \ldots, (x_m, y_m)\}$ of m independent interpretations, the goal is to learn a mapping $h : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{X}$ denotes the set of all instances x_i^k in any interpretation i, with $i \in \{1, \ldots, m\}$ and $\mathcal{Y}$ is the set of target atoms y_i^k. The pair $e^k = (x_i^k, y_i^k)$ is a training example, where $k \in \{1, \ldots, n\}$ and n is the number of training instances in the interpretation i. Each example e^k is, thus, a smaller interpretation, part of the larger point cloud interpretation. Given a new point in a point cloud interpretation we can use h to predict its target category. Next, as explained in Chapter 4, each interpretation x is converted into a bipartite graph G that has a vertex for each ground relation. Figure 7.2(b) shows part of the graph mapped on a point cloud. The graph is the result of grounding the E/R diagram for a particular point cloud.

7.3 Relational Kernel Features

We solve the grasping recognition problem in a supervised learning setting. We employ two variants of the fast neighborhood subgraph pairwise distance

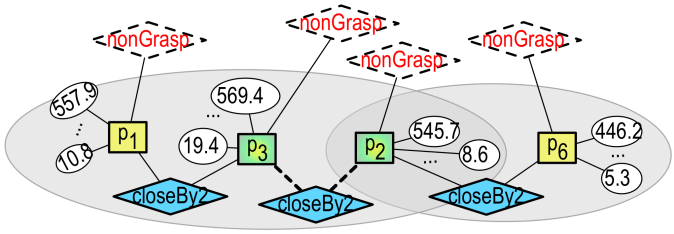


Figure 7.3: From point cloud graph to feature vectors in kLog.

kernel [Costa and De Grave, 2010]. In the graspable point detection problem, our goal is to explore different ways to combine appearance-features that are constrained by spatial proximity in a structural symbolic manner.

As explained in Chapter 4, the kernel is a decomposition kernel [Haussler, 1999] that counts the number of common parts between two graphs. In our case the graph is the contextual information of one point in the cloud. We review the kernel between two graphs as the decomposition kernel defined by relations $R_{r,d}$ for $r = 0, \ldots, R$ and $d = 0, \ldots, D$:

$$K(G, G') = \sum_{r=0}^{R} \sum_{d=0}^{D} \sum_{\substack{A,\, B \in R_{r,d}^{-1}(A,B,G) \\ A',\, B' \in R_{r,d}^{-1}(A',B',G')}} \kappa((A, B), (A', B')) \tag{7.1}$$

where $R_{r,d}^{-1}(A, B, G)$ returns the set of all pairs of neighborhoods (or balls) (A, B) of radius r with roots at distance d that exist in G. The kernel hyper-parameters maximum radius R and the maximum distance D are set experimentally. Figure 7.3 shows a neighborhood-pair feature with $R = 2$ and $D = 2$ for the grasping problem. After we ensure that only neighborhoods centered on the same type of vertex will be compared, constraint imposed by the equation:

$$\kappa((A, B), (A', B')) = \kappa_{root}((A, B), (A', B')) \cdot \kappa_{subgraph}((A, B), (A', B')), \tag{7.2}$$

we solve the grasping problem using two specializations of $\kappa_{subgraph}$. Because we deal both with symbolic and numerical attributed graphs, we employ a hard-soft variant which combines an exact matching kernel for the symbolic relations and a soft match kernel for numerical properties of the relations, and a soft variant which uses only a soft match kernel.

7.3.1 Soft matching

The soft matching kernel uses the idea of multinomial distribution (i.e., histogram) of labels. It discards the structural information inside the graph, however, contextual information is still incorporated by the sum pooling operation applied on the points numerical properties.

$$\kappa_{subgraph}((A,B),(A',B')) = \sum_{\substack{v \in V(A) \cup V(B) \\ v' \in V(A') \cup V(B')}} \mathbf{1}_{\ell(v)=\ell(v')} \kappa_{tuple}(v,v') \tag{7.3}$$

where $V(A)$ is the set of vertices of A and $\ell(v)$ is the label of vertex v. If the atom $point(p_1, f_1, \ldots, f_c, \ldots, f_m)$ is mapped into vertex v, $\ell(v)$ returns the signature name $point$. In this case κ is decomposed in a part that counts the vertices that share the same labels $\ell(v)$ in the neighborhood pair and ensures matches between tuples with the same signature name ($\mathbf{1}_{\ell(v)=\ell(v')}$), and a second part that takes into account the tuple of property values. These are real values and thus, the kernel on the tuple considers each element of the tuple independently with the standard product:

$$\kappa_{tuple}(v,v') = \sum_{c} prop_c(v) \cdot prop_c(v') \tag{7.4}$$

where for the atom $\texttt{point}(p_1, f_1, \ldots, f_c, \ldots, f_m)$, mapped into vertex v, $prop_c(v)$ returns the property value f_c. In words, the kernel will count the number of symbolic labels and will sum property values that belong to vertices with same labels $l(v)$ that are contained in the neighborhood pair.

7.3.2 Hard-soft matching

The hard-soft variant replaces the label $l(v)$ in Equation 7.3 with a relabeling procedure for the discrete signature names. We proceed with a canonical encoding that guarantees that each vertex receives a label that identifies it in the neighborhood graph based on the exact extracted structure of the ball with respect to the relabeled vertex. Then, the exact match kernel for the discrete part is defined as $\kappa_{subgraph}((A,B),(A',B')) = 1$ iff (A,B) and (A',B') are pairs of isomorphic graphs. The isomorphism is ensured by the vertices canonical relabeling. Concerning the real valued properties, we use the standard product as in Equation 7.4 for the tuples of vertices with same relabelings. The spatial

relations injected in the graph and its structure ensures that the pooled features are the ones belonging to vertices with a similar relabeling. In this way, we only sum the features with the same contextual structure. For more details, see [Frasconi et al., 2012].

7.4 Experiments

We address experimentally the benefits of our contribution: the relational kernel-based approach for robot grasping. Specifically, we evaluate whether the relational kernel using contextual symbolic information improve graspable point recognition upon local shape features by investigating the following questions:

(Q1) Does featuring pooling via symbolic relations help solving the robot grasping problem?

(Q2) Does the structural contextual information using the hard-soft matching improve over the soft matching?

(Q3) How do the parameters of the kernel and thus, contextual knowledge, influence the results?

For this purpose we perform experiments with all feature types in turn. We incorporate richer and richer information to assess the importance of pooled features for both kernel variations and validate the best kernel parameters.

7.4.1 Dataset

To investigate the benefit of contextual information via symbolic relations for graspable point recognition we consider a realistic dataset similar to that in [Moreno et al., 2011]. Denoted as S_8, it is gathered using 8 objects: ellipse, rectangle, rounded object, 2 glasses and 3 cups. It contains 2631 instances (1972 positives and 659 negatives). The goal is to evaluate the generalization capabilities of our approach across different object categories based on local appearance features and contextual symbolic ones. We estimate this under the partial view constraint, that is for each object different partial point clouds were gathered representing the object from different view points. The number of views can differ from object to object. Figure 7.4 shows eight views for one of the cups. All views belonging to the same object are mapped to a logical interpretation in our representation. In this way one may consider several views of an object as one interpretation. However, in our practical experiments the

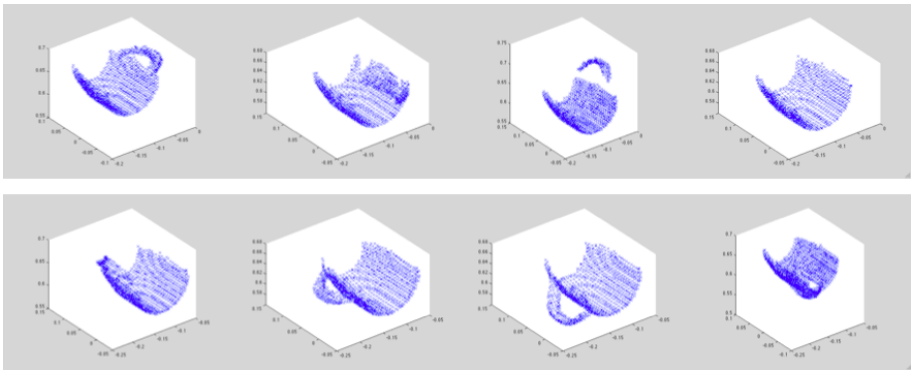


Figure 7.4: Point clouds representing partial views of a cup.

alignment between the views is not realized, such that the contextual information used for each point belongs only to the same view.

7.4.2 Evaluation measures

We apply the one-leave-out validation method for performance evaluation in which one object is used for testing and the rest for training. In all our experiments we have used an SVM with a linear kernel on top of the relational kernel features. The SVM cost parameter was set to 1. We evaluate performance in terms of true positive rate (TPR) and accuracy (Acc) for both datasets. Because the datasets are unbalanced (with more positives than negatives), we also report the area under the ROC curve (AUC) which is not sensitive to the distribution of instances to classes.

7.4.3 Results and discussion

In the following we present quantitative experimental results for all questions. For all results bold font, if present, indicates the best performance. For each feature type we start with local feature vectors and we gradually combine the different relations listed to analyze the impact of the symbolic information. As a baseline for comparison we use the local vectors as sole features, without any symbolic information. We add spatial information by incorporating the user defined symbolic relation `closeBy` as defined in Section 7.2. We report

Shape features	TPR (%) S_8	Acc (%) S_8	AUC S_8
VFH	73.9	59.5	0.46
VFH + closeBy2	92.4	**89.0**	**0.92**
VFH + closeBy3	**97.2**	76.3	0.55
VFH + closeBy2 + closeBy3	94.1	**89.0**	0.91
PFH	72.6	62.3	0.51
PFH + closeBy2	89.7	**87.0**	**0.92**
PFH + closeBy3	**94.5**	74.4	0.53
PFH + closeBy2 + closeBy3	91.3	86.1	0.91
SC	74.3	58.7	0.43
SC + closeBy2	92.7	**89.2**	**0.92**
SC + closeBy3	88.3	69.8	0.54
SC + closeBy2 + closeBy3	**94.1**	88.2	0.91

Table 7.1: Performance results using sphere features. Per object evaluation using hard-soft matching ($R = 2$, $D = 2$).

performance results using the hard-soft matching kernel in Table 7.1 for sphere appearance feature setup and in Table 7.2 for gripper cell setup. Our results show that the use of symbolic relations for features pooling improves the robotic grasping tasks for all shape features used, which answers positively (Q1).

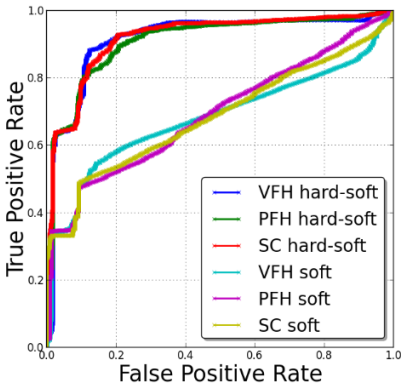
We answer question (Q2) by plotting the ROC curves for both soft and hard-soft kernel for sphere features and hyper-parameters $R = 2$ and $D = 2$. The results in Figure 7.5(a) show that hard-soft matching improves considerably upon the soft kernel. Thus, the contextual structure in the point cloud is highly relevant and ensures the right shape features pooling. In Figure 7.5(b), we fix the shape feature to VFH and the kernel to hard-soft matching and draw the ROC curves for different hyper-parameters of the kernel. The goal is to show how relevant is contextual structure to recognize good grasping points. The best result is obtained for $R = 2$ and $D = 2/D = 4$, which answers question (Q3).

7.5 Related work

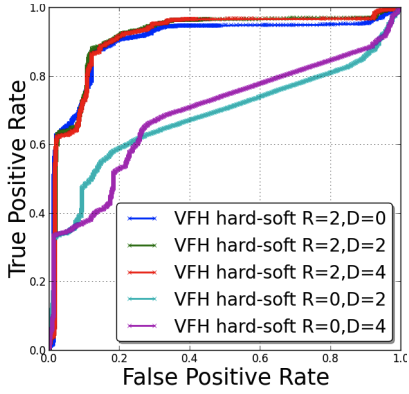
In visual recognition a number of feature extraction techniques based on image descriptors (i.e., SIFT) have been proposed. They usually encode the descriptors over some learned codebook and then summarize the distribution of the codes by a pooling step [Boureau et al., 2010, Jia et al., 2012]. While the coding step produces representations that can be aggregated (or pooled) without losing too much information in the process, pooling gives robustness only to small

Shape features	TPR (%) S_8	Acc (%) S_8	AUC S_8
VFH	70.8	55.7	0.41
VFH + closeBy2	91.3	**88.3**	**0.92**
VFH + closeBy3	**97.2**	76.3	0.55
VFH + closeBy2 + closeBy3	93.0	87.5	0.91
PFH	60.8	50.4	0.40
PFH + closeBy2	90.2	**87.6**	**0.92**
PFH + closeBy3	**97.2**	76.2	0.53
PFH + closeBy2 + closeBy3	91.6	86.7	0.90
SC	75.8	64.0	0.53
SC + closeBy2	90.6	**88.0**	**0.92**
SC + closeBy3	**91.4**	72.7	0.53
SC + closeBy2 + closeBy3	92.4	87.4	0.91

Table 7.2: Performance results using the gripper cell setup. Per object evaluation using hard-soft matching ($R = 2$, $D = 2$).



(a) Soft and hard-soft kernels for $R = 2, D = 2$.



(b) Kernel hyper-parameters for hard-soft matching (VFH feature).

Figure 7.5: ROC curves for the two kernel variants and different hyper-parameters (sphere features, VFH/PFH/SC + closeBy2).

transformations of the image. In standard computer vision tasks one fact that makes the coding step necessary is that descriptors such as SIFT cannot be pooled directly with their neighbors without losing information. Different from this work, our contribution to graspable point recognition considers shape feature pooling without the coding step. To this end, we employ a relational learning technique and show its benefit for robot grasping.

Previous works on visual-dependent robot grasping have shown promising results on learning grasping points from image-based 2D descriptors [Montesano and Lopes, 2009, Saxena et al., 2008a, Saxena et al., 2006]. Other works have learned from combinations of image-based and point cloud-based features [Jiang et al., 2011, Bohg and Kragic, 2010, Erkan et al., 2010]. Saxena et. al. [Saxena et al., 2008a] propose to infer grasping probability from the image filter responses at object points. The projected grasping regions on the image allow to discriminate between graspable and non-graspable points and to perform inference on new objects. However, their model does not consider the parameters of the gripper to estimate the quality of the grasping.

Jiang et. al. [Jiang et al., 2011] extend this approach by computing grasping stability features from the point clouds. Those additional features consider finger's presence, object's symmetry and local planarity. The point cloud features are related to the gripper configuration, while the image-based features are related to the visual graspability of a point. Differently, we consider dense 3D data for both gripper configuration and visual graspability. Kraft et. al. [Kraft et al., 2009, Kraft et al., 2010] propose to learn by exploration the graspable points of an object. Each object has a 3D hierarchical representation based on contours and a 'grasp density', both learned during exploration. The hypotheses tested for the grasp density estimation are reduced by considering groups of contours that define a reaching orientation. Since the learning procedure is attached to each object, scalability issues arise and it is difficult to transfer the skills learned to other objects. A major difference is that we learn shape features that generalize across objects.

Furthermore, a significant number of vision-based grasping methods learn a mapping from 2D/3D features to grasping parameters [Saxena et al., 2008b, Bohg and Kragic, 2010, Montesano and Lopes, 2012, Lenz et al., 2013]. However, it is difficult to link a 3D gripper orientation to local image features without considering contextual or global object information. Only recently, methods that take more global and symbolic information into account have been proposed [Aleotti and Caselli, 2011, Neumann et al., 2012a]. They benefit from increased geometric robustness, which gives advantages with respect to the pre-shape of the robotic hand and general shape of the object, generating more accurate grasps. Nevertheless, global information is exclusively dependent on a complete shape of the object. Object reconstruction based on single views is a difficult problem due to lack of observability of the self-occluded parts and requires several object-related assumptions. Differently, our contribution in graspable point recognition employs a new SRL approach to shape feature pooling that uses kernels on symbolic and numerical attributed graphs to exploit the contextual shape information of objects.

7.6 Conclusions and Future Work

The contribution of this chapter to robot grasping is a statistical relational approach to recognize graspable object points. We represent each point cloud as a graph and we consider symbolic spatial relations between point neighborhood features to exploit extended contextual shape information. Further, we use kernels on graphs to compute discriminative features. We show experimentally that numerical shape featuring pooling via symbolic relations improves robot grasping results upon purely appearance-based approaches. Additionally we show that contextual symbolic structure combined with feature pooling is better than plain sum pooling. The configurable kernel allows us to show the importance of contextual shape information of the object for graspable point recognition based on visual features.

7.6.1 Future Work

As future work we point out three directions. A first direction is to investigate how similar SRL techniques working directly with numerical features can help other robot vision tasks. A second direction is to validate our results on graspable point recognition on datasets that contain a wider range of object categories. Finally, a third direction is to research a more fine-tuned method to semantically mix sub-symbolic (i.e., shape feature vectors) with high-level features (i.e., qualitative spatial relations). Currently, in kLog, the graph kernel implicitly defines a vast set of subgraph features, but it does not allow the user to program this kernel in a declarative way when considering sub-symbolic information. A programmable relational kernel would better exploit and integrate the numerical features.

Part III

SRL for Video Sequence Recognition

Chapter 8

Monitoring Card Games using SRL for Video Sequences

Understanding dynamic scenes of real-world activities from low-level sensor data is essential in many applications. Consider as an example a smart visual surveillance system that is used by rescue robots to recognize traffic collisions, allowing them to act immediately. The more general goal of AI in this context is to create artificial agents that learn by interacting with humans in a natural environment – for instance by observing traffic violation scenarios. In this chapter we tackle the problem of fraud detection. In particular, we are interested in monitoring people playing games and recognizing game related frauds by visual observation in real-world environments. Not only does fraudulent behavior lower the game experience for players, it can also cause economic threats. In fact, the gambling industry generates a large volume of revenues and plays a non-negligible role from an economic point of view. This makes it a driving force behind technical innovation in surveillance systems.

As a first step to be able to recognize game frauds, agents must learn the rules of the game by observing humans playing it. Currently, most computer-controlled agents are trained in virtual environments, where it is assumed that the state of objects is directly available to the agent. Afterwards the agent can use the rules to recognize fraudulent behavior. The difficulty of the tasks is due to several aspects. Firstly, it depends on the richness of the game protocols and the challenge raised by sensor information. Games can be arbitrarily complex due to the number of actions and objects or stochastic aspects. Still, common characteristics between them are their sequential behavior and inherent structure – given by relations between objects, which can elegantly be represented

using *relational sequences*. While complex scenes are best described by high-level, logical representations, video data consist of noisy low-level numerical values. A natural way to incorporate sensor uncertainty from video data is using *probabilistic relational sequences*. They allow us to work with structured terms and, in addition, they capture the inherent uncertainty of object detection. Combining the two types of representation is complex and although this question has been studied before [Tran and Davis, 2008, Needham et al., 2005], there does not yet exist a generally accepted framework that is flexible enough to extract rich symbolic representations from video streams in a general setting.

Secondly, one needs to learn models of dynamic scenes based on such representations to reason about different aspects of the scene. Previous work has learned from sensor data game strategies in a purely logical setting [Needham et al., 2008, Needham et al., 2005, Bennett and Magee, 2007, Fern, 2005]. Even though efficient reasoning about real-world activities requires logical representations, purely logical rules will not suffice due to the inherent noise in video streams. *Relational sequential learning* techniques, however, combine hard logical information with noisy uncertain knowledge. This makes them a good fit for our sequential learning tasks. We first consider the task of extracting the rules of the game. Afterwards we focus on *sequence classification*, that is labelling game sequences as *legal* or *illegal*. Different SRL systems for logical sequences exist [Kersting et al., 2008, Thon et al., 2008]. We employ two of these.

Our first choice is a relational extension of n-grams, called *r-grams* [Landwehr and De Raedt, 2007]. We show experimentally that it is a valid option for simple card games. Furthermore, we propose a theoretical framework to upgrade r-grams to deal with probabilistic observations. Our second choice is a relational extension of CRFs, called *TildeCRF* [Gutmann and Kersting, 2006]. We use it experimentally to solve the same tasks. These two models are representatives of very different classes of learning algorithms. The former is trained using a generative learner, whereas the latter employes a discriminative one.

The chapter is organized as follows. In Section 8.1 we show how to obtain logical and relational descriptions from video streams. In Section 8.2 we formulate the problem setting, explain the sequential learning systems used and present theoretical contributions. Section 8.5 briefly presents related work. Before concluding in Section 8.6, we present our experimental validation in Section 8.3.

Part of the work in this chapter was published in

- Antanas, L., Gutmann, B., Thon, I., Kersting, K., and De Raedt, L. "Combining video and sequential statistical relational techniques to monitor

card games". In Proceedings of the *ICML Workshop on Machine Learning and Games*, 2010.

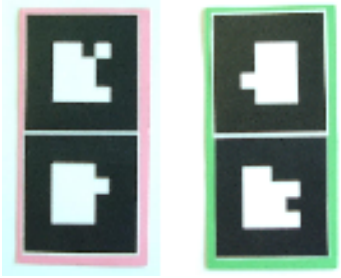
- Antanas, L., Gutmann, B., Thon, I., Kersting, K., and De Raedt, L. "Combining video and sequential statistical relational techniques to monitor card games". In Proceedings of the *Belgian-Dutch Conference on Machine Learning*, 2010.
- Antanas, L., Thon, I., van Otterlo, M., Landwehr, N., and De Raedt, L. "Probabilistic logical sequence learning for video". In *Inductive Logic Programming*, 2009.
- Antanas, L., van Otterlo, M., De Raedt, L., and Thon, I. "Learning probabilistic relational models from sequential video data with applications in table-top and card games". In Proceedings of the *Belgian- Dutch Conference on Machine Learning*, 2009.

8.1 Card Game Video Streams as Relational Sequences

We consider social-interaction scenarios, such as card and board games, which have rich protocols and contain complex spatio-temporal properties. Their complexity can easily be modified by adding new actions and objects or by varying stochastic aspects, which renders such games very suitable for controlled experiments. In this work we consider Uno, a card game which generates simple relational sequences (see Figure 8.1(a)).

Uno is a game for two to seven players. The game objective is to be the first to get rid of all the cards in one's hand to a discard pile. The Uno deck consists of 'common' cards of 4 colors with ranks from `0` to `9` in each color. There are 'action' cards in each color (e.g `skip`) and special action cards or jokers (e.g. `wild`). At any point in time only one exposed card is on the table. Each turn, a player may play a card from its hand that matches either the color or number of the top exposed card, or a (special) action card. Full sequence games may be either *legal* or *illegal*. All moves involving a joker are legal moves, as jokers can be played at any time. A sequence is labeled *illegal* if it contains at least one move which violates the rules. Otherwise it is labelled as *legal*.

(a) Standard Uno cards



(b) Cards with tags *{red, one}* (left); *{one, green}* (right)

Figure 8.1: The UNO game domain

8.1.1 Relational Uno Sequences

We next describe the task of translating videos of Uno games into *relational sequences*. Uno games can be naturally described using sequences of played cards. One major difference in representing sequences is given by the complexity of the underlying language – namely the individual sequence elements. They could be described by sequences of propositional identifiers where each identifier represents a played card (as in Example 30).

***Example* 30.** *A sequence of moves in an Uno game:*
$2-\texttt{red}$, $1-\texttt{red}$, $\texttt{red}-\texttt{draw2}$, $\texttt{wild}$, $\texttt{blue}-6$, $\texttt{blue}-\texttt{skip}$, $\texttt{wild}$, ...

These sequences are propositional and applying statistical models to them requires one to explicitly enumerate all possible states in the game (all possible combinations number-colors). For complex problems propositional representations can lead to a combinatorial explosion in the number of parameters. Instead, we use relational representations in the form of ground atoms describing sequences of elements. Example 31 illustrates a UNO sequence of atoms. This allows one to generalize over similar situations.

Common cards are represented as $\texttt{card}(\texttt{red}, 2)$, and action cards as either $\texttt{card}(\texttt{red}, \texttt{draw2})$ (colored action card) or $\texttt{joker}(\texttt{wild})$ (special action card). Each relational atom in the sequence represents the top exposed card on the discard pile.

***Example* 31.** *The same sequence of moves in a relational form:*

$\texttt{card}(\texttt{red}, 2)$, $\texttt{card}(\texttt{red}, 1)$, $\texttt{card}(\texttt{red}, \texttt{draw2})$,$\texttt{joker}(\texttt{wild})$, $\texttt{card}(\texttt{blue}, 6)$, $\texttt{card}(\texttt{blue}, \texttt{skip})$, $\texttt{joker}(\texttt{wild4})$, ...

We propose a simple and efficient method to visually recognize the cards from video streams by means of tags. We associate with each (previously trained) tag a symbol that represents the object that we want to detect. As an example, a common card contains two tags: one for *color* and one for *number* (action cards have special symbols – e.g. `skip`). In Figure 8.1(b), two different cards with tags are shown together with their associated symbols. We use the ARToolKit framework [Kato et al., 2000] to generate and recognize tags. It uses 2D planar markers and has been employed in augmented reality applications. The introduction of tags enables us to focus on the benefits of SRL techniques instead of focusing on the low-level computer vision tasks. However, the approach is realistic in that similar results can be obtained by applying computer vision techniques for card detection. Several possibilities are available. One can use RGB thresholding for color detection and shape features for digit recognition [Bosch et al., 2007].

Furthermore, the use of tags offers a general framework for symbol detection across different games. Given that with each tag one can associate any symbol, the same set of markers can be used to represent different symbols, depending on the cards of the game (e.g. a marker with associated symbol *one* for Uno can be used to represent symbol *ace* for Poker). A distinction between a game state and a sequence state is useful in the following. A sequence state is obtained for each video frame in time. Typically, there are a number of successive identical sequence states. Such a sequence is called a game state.

In order to obtain the data in the format shown in Example 31 from video streams, a pre-processing phase from tags to logical atoms is required, as described in the following steps.

Step 1: Using ARToolKit, we first obtain a description of each video frame in terms of tags:

$\texttt{tag}(1,2)$, $\texttt{tag}(1,\texttt{red})$, ..., $\texttt{tag}(102,2)$, $\texttt{tag}(102,\texttt{red})$,$\texttt{tag}(103,1)$,
$\texttt{tag}(103,\texttt{red})$,..., $\texttt{tag}(179,1)$,$\texttt{tag}(179,\texttt{red})$, $\texttt{tag}(180,\texttt{red})$, ...,
$\texttt{tag}(186,\texttt{red})$, $\texttt{tag}(187,1)$, $\texttt{tag}(187,\texttt{red})$, ..., $\texttt{tag}(205,4)$,$\texttt{tag}(206,4)$,
$\texttt{tag}(207,\texttt{draw2})$, $\texttt{tag}(207,\texttt{red})$, ..., $\texttt{tag}(242,\texttt{draw2})$, $\texttt{tag}(242,\texttt{red})$, ...

The atom $\texttt{tag}(1,2)$ – for instance – corresponds to observing number `2` in video frame 1. Similarly $\texttt{tag}(1,\texttt{red})$ stands for observing color `red` in frame 1.

Step 2: We compress this sequence by merging tags with the same frame number into one atom. We denote this setting as *uncompressed* data, where each state is a sequence state. Video noise is interleaved between the valid states.

$\texttt{card}(1,\texttt{red},2)$, ..., $\texttt{card}(102,\texttt{red},2)$,$\texttt{card}(103,\texttt{red},1)$,..., $\texttt{card}(179,\texttt{red},1)$,
$\texttt{joker}(180,\texttt{red})$, ...,$\texttt{joker}(186,\texttt{red})$, $\texttt{card}(187,\texttt{red},1)$, ..., $\texttt{joker}(205,4)$,

joker(206, 4),card(207, red, draw2), ..., card(242, red, draw2), ...

Step 3: We replace sets of identical consecutive atoms with one atom. The compressed variant of the sequence above is:

card(red, 2, 102), card(red, 1, 77), joker(red, 7),card(red, 1, 18), joker(4, 2), card(red, draw2, 36), ...

The atom card(red, 2, 102) has as arguments the color, the number (or special action) and the number of identical video frames, respectively. The atom joker(wild, 7) has as arguments the joker symbol and the number of identical video frames. We indicate that the sequence above is a noisy one, where the atoms joker(red, 7) as well as joker(4, 2) are noise. We denote this setting as *compressed* data, where a state in the sequence is a game state. Similar data compression has been done before in [Fern, 2005].

Step 4: We replace the states where the symbols are senseless with unknown tags*. For instance, joker(4, 2) does not make sense as jokers cannot be numbers, therefore it is replaced by joker(unknown, 2). Also, ground atoms such as card(yellow, green, 1) are substituted by card(unknown1, unknown2, 1) since a card cannot contain two colors. After we drop the sequence length, the resulting relational sequence (without the number of identical frames) becomes:

card(red, 2), card(red, 1), joker(unknown), card(red, 1), joker(unknown), card(red, draw2

After pre-processing, the noise-free sequence from Example 31 is in fact the one in Example 32.

***Example* 32.** *'Noisy' relational sequence – the same as in Example 31 – obtained from video streams:*

card(red, 2), card(red, 1), joker(unknown),card(red, 1), card(red, draw2), joker(wild), joker(unknown), card(blue, 6), card(yellow, 6), card(unknown, unknown), card(yellow, 2),card(blue, skip), joker(wild), ...

Tags simplify the recognition task, yet there is uncertainty in the recognition process, due to lighting conditions and occlusion. ARToolKit deals with this by providing confidence values for detected markers. For example, the cards in Figure 8.1(b) are, in fact, recognized in the form: tag(one, 0.8), tag(red, 0.7), tag(green, 0.9), tag(one, 0.8). The attached numbers express the confidence factor with which these tags were recognized. As a consequence, video streams can be represented using *probabilistic relational sequences*. The examples given above only consider markers detected with a confidence factor above 0.5. In our

*ARToolKit introduces inter-tag confusion (e.g. it may recognize the green tag instead of the correct tag 6).

experimental evaluation we consider deterministic atoms. However, we propose a new theoretical framework for the probabilistic case. Although thresholding the confidence removes considerable amount of noise, ARToolKit still introduces non-negligible inter-marker confusion and false positive rates. Added to temporary marker occlusions when cards are manipulated, this translates into a significant source of noise (as shown in Example 32). We address the sequential, relational and noisy aspects of this kind of data by employing sequential SRL techniques.

8.2 Learning Statistical Relational Models from Relational Sequences

There are several learning tasks that can be identified when learning from sequences. One obvious task in the games domain is learning strategies to play the game. We focus on learning various aspects of the dynamics of relational domains from video streams. Afterwards we focus on the task of recognizing fraudulent game sequences. This is done by considering the task of *sequence classification*, that is to label sequences of UNO moves as *legal* or *illegal*. The learning setting is that of learning from interpretations. In this case interpretations are sets of ground atoms sequentially ordered. Each interpretation is a learning example.

8.2.1 R-grams

Relational n-grams or *r-grams* are propositional n-grams [Manning and Schütze, 1999] lifted to logical representations. As explained in Section 2.1.1, n-grams model the joint probability of a sequence $x = \langle x_1 \dots x_m \rangle$ as smoothed Markov chains (a finite mixture of Markov distributions of different orders). An n-gram model is a set of propositional grams which estimates the probability of x as

$$P_n(x) = P(\langle x_1 \dots x_m \rangle) = \prod_{i=1}^{m} P_n\big(x_i | x_{i-n}, \dots, x_{i-1}\big) \tag{8.1}$$

where the conditional probabilities are estimated from a set D of training sequences x using 'gram' counts. Additional smoothing mechanisms which combine grams of different orders prevent the deterioration of model accuracy.

R-grams are obtained by generalizing the sequence elements x_i to *first-order logical atoms*. In our UNO domain a grounded logical atom is, for example,

$x_i = \texttt{card}(\texttt{blue}, 2)$. R-grams exploit the relational structure by considering relational generalizations of grams and by estimating conditional probabilities for non-ground atoms. The generalized atom $\texttt{card}(\texttt{blue}, \texttt{X})$ – for instance – stands for an arbitrary blue card. The probability $P(\texttt{card}(\texttt{blue}, \texttt{X}) \mid \texttt{card}(\texttt{blue}, \texttt{Y}))$ is the probability that a blue card is followed by another blue card.

An r-gram model R of order n is a set of relational grams

$$\left.\begin{array}{l} l_n^1 \\ \ldots \\ l_n^d \end{array}\right\} \longleftarrow l_1 \ldots l_{n-1}$$

where $\forall i : l_n^i$ contains no constant variables and is annotated with probability values $P_r(l_n^i | l_1 \ldots l_{n-1})$ such that $\sum_{i=1}^{d} P_r(l_n^i | l_1 \ldots l_{n-1}) = 1$. Moreover, the heads $l_n^1 \ldots l_n^d$ are mutually exclusive and there are no two grams in R with identical bodies.

By relational generalization, r-grams upgrade n-grams with smoothed probability estimates. This is in contrast to modeling sequences with n-grams by considering all data at the ground level. The basic idea behind smoothing in r-grams is to generalize grams logically, and mix the resulting distributions

$$P_R(x_i | x_{i-n} \ldots x_{i-1}) = \sum_{r \in R} \alpha_r P_r(x_i | x_{i-n} \ldots x_{i-1}) \tag{8.2}$$

where the x_i are logical atoms, R is the set of all generalized relational grams, P_r is the conditional distribution defined by a particular gram and α_r are positive weights with $\sum_r \alpha_r = 1$. Learning an r-gram model from data involves choosing the set R of relational grams, estimating their corresponding probabilities and defining weights α_r for every r-gram r in the selected set. The model is built using maximum likelihood estimates. Similar to n-grams, the r-gram model can consider grams of different orders for an additional smoothing step. A detailed presentation of the r-gram model and learning algorithm is given in [Landwehr and De Raedt, 2007].

Example 33 shows a 2nd order r-gram model. It represents UNO rules extracted from a relational bigram model for the sequence class *legal*. The first two rules show that the next card should have either the same color `A` with probability $P_1 = 0.4$, or the same number `B` with probability $P_2 = 0.51$, while the third shows that a joker can be played next with a probability $P_3 = 0.08$. The last rule models noise. In practice, the bigram model is combined with higher-order models and additional weights are learned for smoothing.

***Example* 33.** *Generalized rules of a relational bigram model for the class* legal.

$$\left.\begin{array}{ll} 0.40 & \texttt{card(C,B)} \\ 0.51 & \texttt{card(A,C)} \\ 0.08 & \texttt{joker(C)} \\ 0.01 & \texttt{card(C,D)} \end{array}\right\} \longleftarrow \texttt{card(A,B)}.$$

Sequence Classification with R-grams

An r-gram model R is built by estimating a model $P_c(x)$ for each sequence class c. An unseen sequence x is labelled by R with the class that maximizes $P(c|x)$. This is equivalent to finding $c \in C$ maximizing $P_c(x) \cdot P(c)$, where $P(c)$ is the prior probability of the class c.

8.2.2 TildeCRF

CRFs [Lafferty et al., 2001] are state-of-the art models for sequence tagging. As in Section 2.1.1, they define a probability distribution $P(y|x)$ as follows

$$P(y|x) = \frac{1}{Z(x)} \exp \sum_{t=1}^{m} F(y_{t-1}, y_t, x) \tag{8.3}$$

where $x = \langle x_1 \dots x_m \rangle$ is the observed sequence, $y = \langle y_1 \dots y_m \rangle$ is the sequence of labels assigned to the observed sequence, $F(y_{t-1}, y_t, x), t \in \{1, \dots, m\}$ is the potential function, and $Z(x)$ is a normalization factor. In the Uno domain, x is the sequence of cards played in one game and y labels every move as *legal* or *illegal*. Sequence classification can result from sequence tagging using several approaches, as described later in Section 8.2.2.

TildeCRF is a relational extension of CRFs where the potential function $F(y_{t-1}, y_t, x)$ is represented as sums of *relational regression trees* [Gutmann and Kersting, 2006]. Relational regression trees upgrade the attribute-value representation within classical regression trees: every test is a logical conjunction of generalized atoms where a variable already introduced in some node cannot appear in its right subtree. This allows for very flexible and compact representations within potential functions.

The compactness and comprehensibility is payed by a more expensive estimation of the potentials, as they are non-parametric functional representations.

TildeCRF[†] follows a gradient tree boosting technique to learn the potential functions [Dietterich et al., 2004]. The resulting potential functions still have the form of a linear combination of features, but the features are complex, i.e., sets of weighted logical rules. Gradient tree boosting is a functional gradient search, where the true gradient is approximated by a regression tree. The gradient, evaluated at all training iterations i, reproduces the potential function as a sum of i regression trees $F(y_{t-1}, y_t, x) = \Delta_1 + \ldots + \Delta_i$. TildeCRF implements several gradient ascent optimizations: plain gradient, plain gradient with line search (LS), conjugate direction boosting [Lutz and Bühlmann, 2006] (CG) and conjugate direction boosting with line search (CG+LS).

A relational regression tree for the UNO domain learned from data is shown in Figure 8.2 [‡]. An inner node represents a generalized atom, a path constitutes a conjunction of generalized atoms, and a leaf represents the regression value of all examples in this leaf. To simplify the evaluation of the gradient at each iteration, TildeCRF does not use the complete input x, but only windows $w_t(x)$ of fixed size d. Thus, the potential has the form $F^{y_t}(y_{t-1}, w_t(x))$. The sliding window w_t is implicit in the predicate `card/5` and `joker_played/4` as they take the offset from the current position `Pos` as input argument. One advantage of using a relational tree representation is, that this window does not need to be constructed explicitly. That means, the input data does not have to be reformatted when – for instance – changing the window size. Instead, one specifies the window size d implicitly during learning by means of background knowledge. In our example, the third argument for `card/5` specifies the offset from the current position `Pos`. Therefore, by varying that constant from $-d$ to d one obtains the same effect as preprocessing the input data towards a fixed window size.

Consider

$$0.495 \ : \ \texttt{PreviousLabel} = \texttt{legal}, \ \texttt{card}(\texttt{Pos}, \texttt{ID}, -1, \texttt{A}, \texttt{B}),$$

$$\texttt{card}(\texttt{Pos}, \texttt{ID}, 0, \texttt{A}, \texttt{C}), \ \texttt{card}(\texttt{Pos}, \texttt{ID}, 1, \texttt{A}, \texttt{D})$$

taken from the regression tree. It groups all ground instances, where `A` is substituted by some term such as {`A`/`red`} or {`A`/`green`}. The meaning of this rule is, that within 3 consecutive rounds a card with color `A` has been played and so far, nobody has cheated. The free variables `PreviousLabel`, `Pos`, and `ID` represent the previous label y_{t-1}, t, and the identifier of the input sequence x, respectively. Relational abstraction makes useful predictions possible in very

[†]Available at `http://www-kd.iai.uni-bonn.de/index.php?page=software_details&id=17`.

[‡]The predicate `card_played` in the figure is marked as `card` in the text.

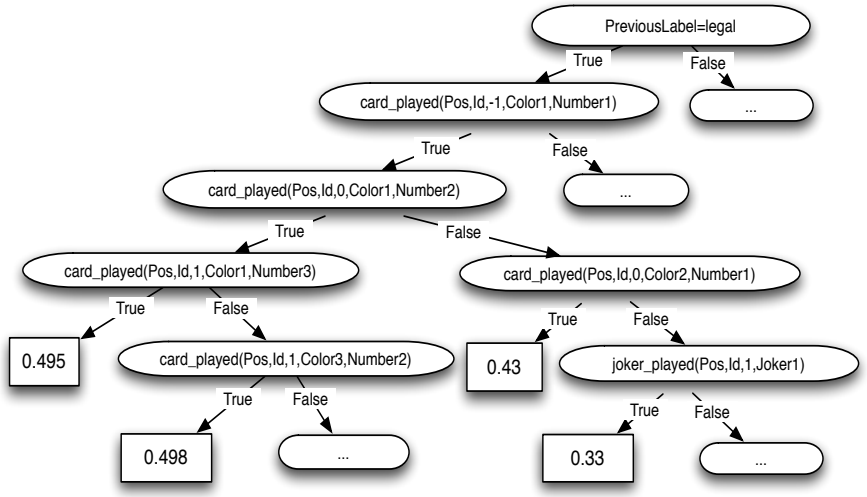


Figure 8.2: A learned regression tree by TildeCRF representing the gradient in the first iteration. Internal nodes represent tests – queries in Prolog form – and leaves represent the output. Parts of the tree have been removed due to space restrictions (indicated by ...).

large state spaces, where many of the states are never observed in the training data. Example 34 illustrates an instantiation of this rule.

***Example* 34.** *In our example a variable assignment* $\{\texttt{ID} \leftarrow 1,\ \texttt{Pos} \leftarrow 2,\ \texttt{PreviousLabel} \leftarrow \texttt{legal}\}$ *together with the set of ground atoms*

$$0.495\ :\ \texttt{PreviousLabel} = \texttt{legal},\ \texttt{card}(1, 1, \texttt{red}, 1),$$

$$\texttt{card}(2, 1, \texttt{red}, 4),\ \texttt{card}(3, 1, \texttt{red}, 8)$$

is sorted into the leftmost leaf, i.e., the value 0.495 *is assigned. In contrast, changing the third atom to* `card(3,1,green,4)` *yields* 0.498 *as value.*

Sequence Classification with TildeCRF

There are several ways to get a sequence classifier from trained CRFs (i.e., to predict a single label for the entire sequence). In order to do so, CRFs first have to be trained accordingly: for all sequences x belonging to class c one provides

a training example of the form (x, y) where $y = \langle c \dots c \rangle$ is a sequence of the same length as x.

The most simple approach – similar to r-grams – is to calculate the likelihood $P(y|x)$ for the label sequence $y = \langle c \dots c \rangle$. The predicted class is the one with the highest likelihood. We refer to this as *global vote*.

$$P(y|x) = \arg\max_{c \in C} P(\langle c \dots c \rangle | x) \tag{8.4}$$

The advantage of this approach lies in its simplicity, as it does not require to calculate the normalization factor $Z(x)$.

Another option is to predict every atom y_t in the output sequence individually. This makes sense when we want to maximize the number of correctly tagged input atoms. To do so, one first predicts a sequence of class labels y either by using the Viterbi (Vi) tagger or the forward-backward (FB) tagger. In a second step, we use the *majority vote* classifier. It treats each position as an equally important vote for a class. In this case, we count the number of appearances of each class labels $count(c, y) := |\{t \in \{1, \dots, m\} \mid y_t = c\}|$. Then, the sequence x is assigned to class c with probability

$$P(c|x) = \frac{1}{m} count(c, H(x)) \tag{8.5}$$

We will refer to the two version of this classifier by *Vi majority* and *FB majority*, respectively.

Finally, for our binary fraud detection problem, we additionally consider a third classifier. We predict the sequence as illegal, if there is at least one position labeled as illegal. We refer to this as single rule mode. Similarly to the majority vote, one first has to tag the whole sequence. Rule mode can be also combined with forward backward and Viterbi. We will refer to each version of this classifier as *Vi rule* and *FB rule*, respectively.

8.3 Experiments

The purpose of the experimental section is to investigate how TildeCRF and r-grams for logical sequences perform on the different Uno datasets when employed for sequence classification (or fraud detection) and if they can extract

the rules of the game. We also examine how they compare to propositional approaches. To this aim, we investigated the following questions:

(Q1) Do r-grams perform well when dealing with real-world video data?

(Q2) Does TildeCRF outperform propositional CRFs?

(Q3) Can TildeCRF be used for sequence classification tasks although it is trained for tagging?

(Q4) Is the majority vote a good classifier when used with TildeCRF?

(Q5) Which of the two SRL approaches works better?

Answering (Q1) and (Q2) aims at investigating the additional flexibility of relational models for (relational) sequences. (Q3) checks to which extent a model that was designed for tagging can be used for classification. (Q5) evaluates the majority vote classifier when used with TildeCRF, compared to the other classification approaches used. Finally, (Q4) compares TildeCRF with r-grams. Before we describe the experiments carried out and discuss their results, we present the datasets and the evaluation metrics considered.

8.3.1 Datasets

Experimental data was collected from video sequences of people playing the game with the special tagged cards, using a subset of the UNO cards (without the doubles). The data was gathered with a camera mounted on the ceiling so that it captured the playing deck at any moment. The illegal games were played by 2 players – a fair player and a fraudulent one, while the legal ones by 2 honest players. We shall call this dataset UNO_{Real}. Although the use of tags simplifies the recognition task, ARToolKit introduces non-negligible inter-marker confusion and false positive rates. Added to temporary marker occlusions when cards are manipulated, this translates into a significant source of noise. In addition to the real-world dataset, we generated simulated UNO games. The reason was to have better control over different parameters and to examine the influence of noise on the results. We obtained the datasets UNO_0, UNO_{15}, UNO_{30}, UNO_{50}, where the subscript represents the level of artificially generated noise, as described in Appendix A.

Each dataset contains 50 game sequences with an even distribution of the two classes. We used stratified 5-fold cross validation on these datasets. The folds were built by randomly assigning the examples to folds such that the number of legal and illegal examples are evenly distributed. For both legal and illegal

examples we randomly sampled from examples with high and low level of noise and, in the case of illegal examples, we sampled from the distribution of the low and high number of incorrect moves per sequence, while in the case of legal examples from the distribution of the low and high sequence lengths. The absence of such a stratification can give an uneven distribution of noisy, low-level illegal examples and noise free, high-level illegal examples, which results in a standard deviation often higher than 10%.

8.3.2 Evaluation Metrics

We use the *accuracy* per instance to evaluate the classification performance for both r-grams and TildeCRFs experiments. Accuracy reports the fraction of test sequences classified correctly. Additionally, we note down the *log-likelihood* for TildeCRF. Log-likelihood (llh) reports the normalized conditional log-likelihood of test sequences as $n^{-1} \sum_i^n P(y^i|x^i)$, where $y^i = \langle c, \ldots, c \rangle$ and n is the number of test sequence instances.

8.3.3 Results

A set of experiments with r-grams answers question (Q1). We used the UNO_{Real} dataset and 1 up to 4 r-gram lengths were considered. We report results both with the compressed and uncompressed settings. For r-grams we trained two models, one for each of the classes *legal* and *illegal*. We used both models to classify a sequence as described in Section 8.2.1. The 5-fold cross-validation accuracy was averaged for each r-gram length.

As shown in Figure 8.3 for compressed, 'clean' data, the r-grams are able to best learn with a bigram model, while for compressed, noisy data a trigram model gives a better accuracy. For the uncompressed data a bigram model performs not as good, which can be explained by the fact that in an uncompressed sequence two consecutive ground atoms usually describe the same game state, whereas in the compressed version they describe two different game states. Note that for the uncompressed data the model is best learned also by trigrams. Example 33 illustrates a bigram model built from data, in the 'clean' setting. It shows that r-grams can learn the rules of the UNO game. The first two rules show clearly that the next card should have either the same color or the same number. While in a relational setting the rules are very compact by using variables, in a propositional one these rules are fully grounded.

To answer question (Q2) we consider all datasets. For TildeCRF we used the classifiers described in Section 8.2.2. A comparison between the relational CRF

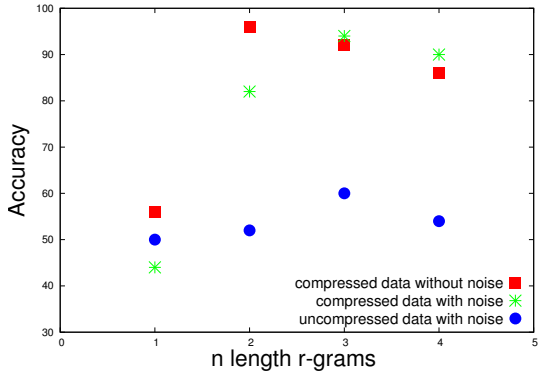


Figure 8.3: Accuracy for different r-gram lengths (UNO$_{Real}$).

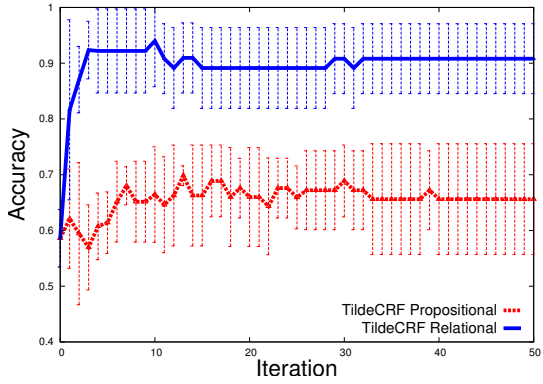


Figure 8.4: Performance of TildeCRF on UNO$_{Real}$. With a relational language bias TildeCRF outperforms the propositional setting. The plain gradient optimization and the Vi majority classifier were used for this experiment.

and its propositional version for UNO$_{Real}$ is shown in Figure 8.4. The accuracy of predicted sequence classes after each training iteration is reported. The relational version outperforms consistently the propositional version, obtaining a test set performance of 91%.

In Figure 8.5 experimental results with relational CRFs for all optimization methods are summarized on the left. The ones for propositional CRFs are on the right. We can see that in all cases, for either accuracy or log-likelihood, the relational version gives consistently improved results, on the same scale and the same number of iterations. This answers affirmatively (Q2). The model trained

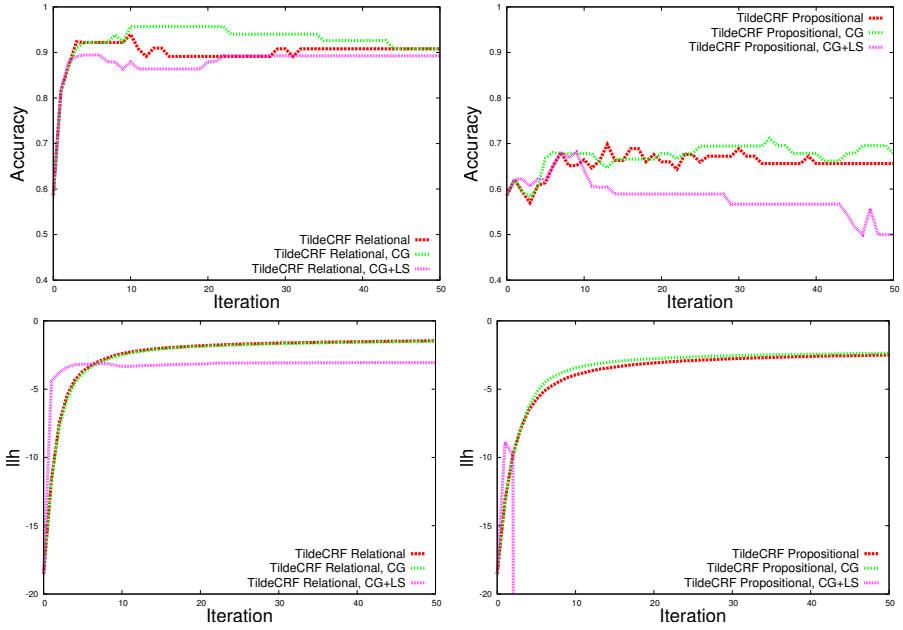


Figure 8.5: Performance on UNO$_{Real}$ with Vi majority: relational (left) vs propositional (right) CRFs.

using a propositional language bias shows poor performance due to overfitting. The learner tries to fit each possible combination of color and number to another legal combination color and number or a joker. A part of the regression tree learned by TildeCRF is shown in Figure 8.2. It presents some of the UNO rules, saying that the next card should match the card color, the card number or should be a joker. While in a relational setting the learnt rule can be nicely expressed using variables, in a propositional one such a tree explodes, by trying to learn each possible combination for which the rule holds. For UNO$_{Real}$ the conjugated gradient achieves an accuracy of 96%, slightly better than the plain gradient (92%). In terms of log-likelihood the performance of the two methods are basically the same.

The performance of the conjugated gradient for all levels of noise is illustrated in Fig. 8.6. One can notice that the accuracy of TildeCRF decreases along with the increase of noise level. The results for real-world dataset in terms of accuracy are situated in between the results for UNO_{30} and UNO_{50}, which motivates us to believe that the real-world noise level is in the same range of values.

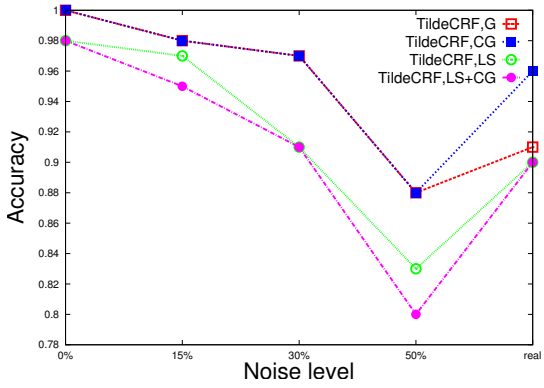


Figure 8.6: All datasets; Influence of the noise on accuracy performance for all methods; Classification method: Viterbi majority; 5-fold crossvalidated

In order to tackle question (Q4), results for all classification approaches: Vi majority, Vi rule, FB majority, FB rule and global label are described in Table 8.1. In general all the sequence classification criteria can be used for the classification task, except FB rule vote, which gives the poorest results and a high standard deviation for one of the dataset. This answers affirmatively question (Q4) and lets us believe that it is a good classifier when used with TildeCRF. Viterbi majority vote gives the best performance for all datasets.

When building the TildeCRF model we allow regression trees up to 12 leaves and a learning rate $\mu = 1$. As a stopping criteria for our results we monitored the validation set and reported the test accuracy corresponding to the second validation accuracy before it decreases a few times consecutively. A sliding 9-window is used.

Criteria	UNO$_0$	UNO$_0$	UNO$_0$	UNO$_0$	UNO$_{Real}$
FB majority	$\mathbf{1.00 \pm 0.00}$	0.97 ± 0.04	$\mathbf{0.97 \pm 0.05}$	$\mathbf{0.88 \pm 0.05}$	$\mathbf{0.96 \pm 0.06}$
FB rule	0.92 ± 0.08	0.89 ± 0.24	0.84 ± 0.07	0.80 ± 0.07	0.87 ± 0.09
Vi majority	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.98 \pm 0.03}$	$\mathbf{0.97 \pm 0.05}$	$\mathbf{0.88 \pm 0.08}$	$\mathbf{0.96 \pm 0.06}$
Vi rule	0.97 ± 0.06	0.96 ± 0.06	$\mathbf{0.97 \pm 0.05}$	0.85 ± 0.09	0.92 ± 0.07
Global	$\mathbf{1.00 \pm 0.00}$	0.96 ± 0.05	0.94 ± 0.05	0.84 ± 0.09	0.90 ± 0.07

Table 8.1: Performance of TildeCRF (conjugated gradient) on all UNO datasets using the defined classification approaches. The bold notation shows the best accuracy scores.

Model	Setting	Accuracy
r-grams	Length 2	0.84 ± 0.12
	Length 3	$\mathbf{0.94 \pm 0.05}$
	Length 4	$\mathbf{0.94 \pm 0.05}$
	Length 5	0.92 ± 0.04
TildeCRF	Vi majority	$\mathbf{0.96 \pm 0.06}$
	Vi rule	0.92 ± 0.07
	FB majority	$\mathbf{0.96 \pm 0.06}$
	FB rule	0.87 ± 0.09
	Global label	0.90 ± 0.07

Table 8.2: Classification results for UNO_{Real}. The bold notation shows the best accuracy scores.

Comparative results between r-grams and TildeCRF are shown in Table 8.2. Both systems perform well and obtain competitive results on the sequence classification task with respect to the predicted accuracy. However, the advantage of the generative model is that it is easier to understand and, for the UNO domain, also faster to train. To allow for a true comparison of the results, the same folds were used for cross-validation when evaluating the models. Nevertheless, there are small representation differences of the sequences in order to optimize the performance of each method. This answers question (Q5).

8.4 Augmented r-grams

The original r-grams have been developed for working with sequences $x = \langle x_1 \dots x_m \rangle$ that are certain (each element x_i is known to be true). In the real-world, however, each element has a degree of uncertainty. Thus, we introduce an extension of r-grams that deals with sequences where each observation x_i is estimated to be correct with probability p_i. In a more general setting each x_i is a probability distribution over possible tag symbols or classes introduced by object detection. In our case the observations are the defined symbols associated to previously trained tags. Example 35 shows such a probabilistic sequence.

***Example* 35.** *A sequence example where each element is a probability distribution over possible tag symbols*

$\texttt{card}([one : 0.8, two : 0.0, red : 0.1, blue : 0.1], [one : 0.0, two : 0.0, red : 1.0, blue : 0.0]),$
$\texttt{card}([one : 0.7, two : 0.1, red : 0.2, blue : 0.0], [one : 0.0, two : 0.0, red : 0.1, blue : 0.9]),$
$\texttt{card}([one : 0.2, two : 0.8, red : 0.0, blue : 0.0], [one : 0.0, two : 0.2, red : 0.0, blue : 0.8]),$

where the defined set of observations is $O = \{one, two, red, blue\}$. *The atom* $\texttt{card}([one : 0.8, two : 0.0, red : 0.1, blue : 0.1], [one : 0.0, two : 0.0, red : 1.0, blue : 0.0])$ *reflects the degree of uncertainty of the atom* $\texttt{card}(one, red)$ *(previoulsly assumed certain). The card has the symbol one as its number observation with probability* 0.8. *Similarly, the card has the symbol red as its number with probability* 0.1*, the card has red as its color observation with probability* 1.0 *and so on.*

In order to deal with such *sequences of probabilistic relational atoms*, we propose *augmented r-grams* or *ar-grams*. The new model extends the original r-grams by introducing a new type of smoothing due to the probabilities in the sequences. Therefore it requires new methods to (i) evaluate the conditional probabilities and (ii) estimate the r-gram probabilities. The probability of an observed sequence o using ar-grams is defined as

$$P_n(o) = \prod_{i=1}^{m} P_n\big(x_{o_i} | x_{o_{i-n}}, \ldots, x_{o_{i-1}}\big) = \tag{8.6}$$

$$= \prod_{i=1}^{m} \sum_{k=1}^{|\mathcal{S}|} P_n\big(x^k_{o_i} | x^k_{o_{i-n}}, \ldots, x^k_{o_{i-1}}\big) \cdot P\big(x^k_{o_{i-n}} | o_{i-n}\big) \cdot \cdots \cdot P\big(x^k_{o_i} | o_i\big) \tag{8.7}$$

where o is the sequence of probabilistic atoms, $\mathcal{S}$ is the set of all possible sequences $x^k_o = x^k_{o_{i-n}}, \ldots, x^k_{o_i}$ of length n that can be observed, starting from the set of possible observations O. $P_n(x_{o_i} | x_{o_{i-n}}, \ldots, x_{o_{i-1}})$ is a standard (smoothed) r-gram model as defined by Eq. 8.2. $P(x_{o_i} | o_i)$ is the probability of x_{o_i} given the observation o_i. There is always one probability for each observation atom in the sequence x^k_o. It can be estimated by multiplying the probabilities of individual symbols involved in the relational atom. We also assume accurate estimates for the observations at each video frame, which makes the observation of every sequence element independent (see Eq. 8.7). The model can be built using maximum likelihood methods. Dealing with sequences as described above makes our setting different from the standard hidden data problems[§]. We do not have noise observation, as in the case of hidden data models, but distributions over possible expected true observations.

[§]When models such as Logical Hidden Markov Models [Kersting et al., 2006] can be used.

8.5 Related work

We present the related work along two axes. The first axis is the use of sequential relational models to solve problems in other domains. For example the relational extension of HMMs, called LoHMMs, was employed for two bioinformatics problems [Kersting et al., 2006], while relational sequence alignments were used for information extraction in medical texts and in protein fold description [Karwath and Kersting, 2007]. R-grams were successfully applied before to Unix user modeling, protein fold prediction, and mobile phone user pattern analysis [Landwehr and De Raedt, 2007]. Finally, TildeCRF was used to solve protein fold classification problems [Gutmann and Kersting, 2006].

The second axis is the use of sequential models to analyze video data. Very related is the work in [Needham et al., 2005], [Santos et al., 2006], [Needham et al., 2008] where similar relational sequences were obtained from video and audio data by clustering extracted video features. However, one disadvantage is that feature clustering can give much redundancy and objects can easily be misclassified. A second disadvantage is the use of a purely relational technique to learn the rules of the game, without considering the statistical aspect of noisy perceptual information.

Other papers continued our work published in [Antanas et al., 2009], [Antanas et al., 2010a], [Antanas et al., 2010b]. One extension makes use of probabilistic relational input [De Raedt and Thon, 2010], however their approach is different than ar-grams. Other extensions look at more complex games with richer game states [Barbu et al., 2010], [Hazarika and Bhowmick, 2012]. A major difference is that they do not employ statistical relational techniques as we do. Furthermore, [Kaiser, 2012] presents an integrated vision and robotic system that plays and learns to play simple physically-instantiated board games that are variants of games such as Tic Tac Toe or Hexapawn. In a broader context the work on human activity recognition is also related. Several relational graph mining [Sridhar et al., 2010a] or inductive logic programming [Dubba et al., 2010] techniques have been employed to learn activity event models with more complex states. Differently, we focus on learning from sequences of relational atoms.

8.6 Conclusions and Future Work

We presented a method to obtain relational descriptions from video streams. Starting from these descriptions, we successfully employ r-grams and TildeCRF models to show that the two SRL approaches can learn the dynamics of the

UNO game and perform well in detecting fraudulent games. We motivated that a more powerful r-gram model is needed to explicitly incorporate the inherent uncertain aspects of video streams. In this direction we propose a novel extension of the r-grams, called ar-grams. Finally, we provide consistent experimental evidence that shows the benefits of using relational representations over propositional ones.

There are several useful observations with respect to the experimental evaluation. They concern the representation part, that is bridging the gap between low-level features and high-level logical representations. First, although tags eased the pre-processing steps, finding a good representation with respect to the SRL systems required several tests, e.g., using compressed/uncompressed sequences, replacing the senseless detections with *unknown* symbols or taking into account discretized lengths of game states (in the case of r-grams). Thus, building a general purpose computer vision framework that allows for efficient pre-processing steps to get from a continous format (i.e., the image) to a symbolic one (e.g., the tags) would allow more resource allocation to the representational aspect. Second, there were several advantages of the generative method over the discriminative one. R-grams required less background knowledge, were faster to train, provided more understandable models and had similar performance to the discriminative approach, although used in a setting with limited amounts of data.

8.6.1 Future Work

As future work there are several possible directions. An obvious next step is to use real games and computer vision techniques for symbol detection instead of markers. Another possibility is to employ ar-grams experimentally to evaluate how their performance compares to the one when using deterministic input. Moreover, an interesting idea is to address the problem of recognizing less obvious fraudulent behaviours for games with richer protocols. This involves also more complex models, where each game state corresponds to a full interpretation. In this case, however, compressing similar sequence states will pose difficulties, as finding similar interpretations via interpretation matching is NP-hard. Thus, the SRL extension should be able to work with uncompressed sequences. In the generative case this issue could be approached, for example, by extending ar-grams to *m-skip ar-grams* [Guthrie et al., 2006]. Finally, although the employed SRL methods work well on our datasets, it would be interesting to see whether other methods, such as LoHMMs, perform as well.

Finale

Chapter 9

Summary and Future Work

This chapter summarizes the contributions of the thesis, formulates conclusions by answering the questions addressed in the introduction and discusses main directions for future work.

9.1 Thesis Summary

This thesis has investigated the use of several SRL techniques for different real-world visual recognition problems. The resulting systems are feasible with respect to the visual recognition problems addressed and have shown promising experimental results. In particular, we have explored relational representations in five different settings: 1) object recognition, 2) scene classification, 3) robotic pre-grasp prediction, 4) grasping point classification, and 5) sequence classification. We now briefly summarize the key contributions and experimental conclusions.

Part I of the thesis was devoted to *relational scene understanding*. Several SRL approaches were proposed to tackle different recognition problems: object recognition, scene recognition and scene understanding. In contrast to statistical scene understanding, relational scene understanding uses symbolic relations and relational background knowledge for visual recognition, leading to declarative and more intuitive recognition systems. In contrast with pure appearance and statistical approaches, they take into account qualitative spatial relations and sparser appearance cues. In Part I we first reinvestigated some old ideas from 1970s and 1980s, starting from modern low-and mid-level appearance

features. Then, in Chapter 4, we explored two new relational approaches to understand hierarchically images of houses. The goal there was to recognize constituent doors, windows and houses at different levels of semantic granularity. The first approach was based on a relational distance and combined feature extraction, qualitative spatial relations, and compositional hierarchies in one framework. The second approach extended the first one, by replacing the relational distance with a kernel for relational structures. We showed empirically the interplay between structural and appearance-based aspects and good detection results thanks to the use of relational representations and the flexible declarative language, although sparser appearance cues were used. Furthermore, in Chapter 5, we approached the scene understanding problem via two sub-problems, that is object recognition for indoor scenes, and scene classification for both indoor and outdoor scenes. The experimental results, competitive with state-of-the-art using sparser cues, show the benefits of relational representations for both visual sub-problems. Furthermore, we successfully distinguished between indoor scene categories with the help of relations in cases where non-relational systems could not. Overall, our results show the feasibility and effectiveness of relational formulations for specific visual recognition problems by combining relational knowledge representations with computational vision.

Part II introduced *relational recognition for robot grasping*. Specifically, we proposed a probabilistic logic pipeline that leverages world knowledge, in the form of object/task ontologies, and low-level data, in the form of point clouds for robot grasping. State-of-the-art approaches learn a function that maps directly visual perceptions to good grasps. Chapter 6 upgrades robotic setups proposed in the literature in two ways. First, we introduced a probabilistic logic module which can semantically reason about the object category, suitable tasks and pre-grasp configurations. Experimentally, our pipeline showed robustness to uncertainty and missing information and confirmed the importance of high-level reasoning for robot grasping, which is most conveniently expressed using relations/logic. Using probabilistic logic reasoning we could improve over local shape features on both simulated and real robot platforms. Second, we proposed a relational kernel-based approach to map part-related shape features to good grasping points. Our experiments showed that numerical shape feature pooling via symbolic relations improves robot grasping results upon purely appearance-based approaches. This confirms the benefits of contextual knowledge for graspable point prediction by means of relational representations.

Part III explored two different *probabilistic relational sequential learning techniques for video streams of dynamic game scenes*. We considered the tasks of extracting the rules of the game and sequence classification in Chapter 8. We provided consistent experimental evidence to support the benefits of using relational representations over propositional ones when detecting fraudulent

behaviour from video in games involving sequences of logical atoms.

9.2 Discussion

Relational visual recognition aims at solving visual recognition problems by employing relational representations. It enjoyed popularity in the early vision work, where relational and logical languages, sometimes in declarative and graph-like forms, have been employed to solve different computer vision problems. The interest in such representational schemes was based on the intuition that people are interpreting visual scenes in terms of high-level symbolic knowledge, such as component objects, their complex structure, semantic configuration and interaction. As examples, consider the tasks of understanding and recognizing individual house facades, distinguishing between restaurant and bar scenes, or finding the best robot grasp based on the object configuration and task-related constraints. In these cases, scenes are best described in terms of relational languages or, equivalently, (higher-order) graphs. Moreover, this approach was convenient at that time given the limitations of hardware, data, scientific technologies and low-level vision routines. Today, however, these elements have become much more mature. Still, relational visual recognition is currently an inactive area of research as the computer vision community takes a leading statistical position. This is also due to the pure symbolic nature of relational representations, which partly limited progress in early vision. Nevertheless, recent successes in combining symbolic representations with statistical principles and the maturity of the aforementioned resources, motivated us to revive old ideas and to reinvestigate them by means of SRL. Starting from low-and mid-level visual solutions and building on top of them, we contributed with several developments which answer the main research question of this thesis introduced in Chapter 1. We can have real-world relational visual recognition systems and the problems tackled by these systems can use several advantages of SRL. These insights give the perspective of moving towards more general and effective visual recognition systems with the help of SRL.

We showed that visual recognition problems can benefit in several ways from (statistical) relational learning. These gains depend on the application and the problem at hand. First, relational representations are more intuitive to understand and can describe domains exhibiting considerable structure using qualitative relations. In Chapters 4 and 5 we demonstrated that they can declaratively express the composition of scenes into objects, parts of objects and lower-level substructures and the symbolic dependencies among these parts. For example, we showed how typical houses can be described as consisting of spatially aligned windows and doors. Thus, relational approaches offer a

flexible and interpretable way to consider spatial, functional and contextual information in visual scenes. Second, they employ structured input features and may output a structured explanation of the scene (see Chapter 4). Third, when grounded in statistical learning frameworks, i.e., kernel-based methods exemplified in Chapters 4 and 5, relational approaches show more robustness to noise, provide improved results, and are computationally more tractable. Fourth, it can be that appearance features are weaker discriminative cues and it is their consistent and complex semantic interaction that helps solving the recognition problem. In these cases relational representations can provide a principled way to represent exact metric locations as arbitrarily higher-order relations among the component objects of a scene. For example, one can describe a bar scene as having "a variable number of chairs of similar size, close to each other and aligned horizontally along a counter". This is a more expressive representation than a fixed grid and more robust than continuous locations. Chapter 5 illustrated this aspect. Finally, domain and world knowledge can easily be incorporated using logical rules. In Chapters 4 and 5 spatial theory was specified in an intuitive way for image understanding. In robot manipulation, high-level reasoning about the object, its properties, task and grasp constraints, object/task ontologies and object-task affordances, can be best performed using compact rule-based logical models. Furthermore, SRL provides probabilistically enhanced logical models to reason about the uncertainty in the perceived world. We exemplified these aspects in Chapter 6. The logic-based representation can naturally generalize over similar object parts and object/task categories. This allowed us to experiment with a wide range of object and task categories, moving visual recognition towards more general systems.

Related (statistical) relational visual recognition work includes several old frameworks for general scene understanding [Tenenbaum et al., 1975, Hanson and Riseman, 1978, Shapiro, 1983], geographical region interpretation [Reiter and Mackworth, 1989] or aerial image understanding [Matsuyama and Hwang, 1985]. Currently, there are, however, only few recent works that actually start from automatically extracted image primitives and employ relational learning techniques. They tackle problems such as event recognition for parking lots [Tran and Davis, 2008] or airport aprons [Dubba et al., 2010], object recognition for robot grasping [Marton et al., 2009] and navigation [Farid and Sammut, 2012], or understanding scenes of buildings [Hartz et al., 2009, Xu and Petrou, 2009]. These systems are also mainly problem focused and their impact is not enough appreciated in the visual recognition community. Nevertheless, the aforementioned work and our contributions in relational visual recognition are good starting points for more general systems. The development of general, complete and effective relational visual recognition can be identified today as a promising, but also difficult direction for relational learning and vision.

9.2.1 General Remarks and Take Away Messages

To summarize, the thesis has contributed with the use of several SRL techniques for different visual recognition problems. These contributions should act as evidence that extending current statistical approaches to visual recognition with relational representations is feasible in practice for real-world problems. The relational approaches proposed in this thesis are characterised by expressivity, but also generality. The later is achieved via the background knowledge, relational language and learning methods employed. For example, kLog's language was used to tackle two different tasks in Chapters 4 and 5. Nevertheless, these characteristics were achieved by focusing on particular problems and application domains. For instance, in Chapter 4 the relational features were tailored to the house facade domain. While the language and the learning method can be employed for other structured domains as well (e.g., human body recognition), changes with respect to the primitives and the spatial relations used – thus, feature construction, are required to obtain good results. For example, for the house facade domain, we started from 2AS primitives and used several spatial relations among them to recognize windows and doors. However, we could also start directly from window and door detections obtained using purely appearance-based detectors and employ spatial relational context to improve them. Finding the semantic level where relational representations can help the most is problem-dependent and open for investigation. For the indoor scene domain we used more general primitives, i.e., visual words and semantic objects, however, the feature engineering was still problem-dependent.

One should also keep in mind that there must be a trade-off between the generality, expressivity, efficiency, and performance of relational approaches to visual recognition problems. High generality may involve less discriminative features and may not pay off performance-wise when compared to purely statistical approaches which are computationally faster. Finding the right discriminative relations involves good knowledge of the domain and pay off only for highly structured domains or where relational context plays a relevant and consistent role. Furthermore, employing relational representations for unstructured problems or where contextual information is unavailable (e.g., images focused on certain object categories) is not beneficial. For example, one can use relational representations to obtain spatial pyramid features from an image representing a cat. However, if no relational context is available to justify this choice, using a fancy representation will not bring any additional benefits for cat recognition. In this case employing low-and mid-level visual routines already available is more convenient.

The aspects presented above are tightly related to combining visual recognition and SRL techniques. This combination implies filling the semantic gap

between the low-and mid-level features and relational symbolic representations, or, in other words, combining purely statistically methods with relational representations for visual recognition problems, such that advantages of relations and logic can be used at full potential. Practically, for the work done in this thesis, it was rather challenging to bridge the semantic gap and to incorporate symbolic relations expressing world knowledge into the real-world tasks tackled. Filling this gap highly depends on the domain and the targeted visual problem. This implies that extending this work asks for several extra developments currently unavailable in the SRL frameworks employed. A first one is a principled way to semantically mix sub-symbolic (e.g., shape feature vectors) with high-level features (e.g., qualitative spatial relations). To meet the problem-depend needs, SRL frameworks should also integrate several well-established low and mid-level visual cues or primitives for fast declarative experimentation. Another requirement is the possibility to specify the features of interest in a flexible and declarative manner. Because our goal is to combine numerical and symbolic features, kernel methods continue to be a promising choice. Currently, in kLog, the graph kernel implicitly defines a vast set of subgraph features, but it does not allow the user to program this kernel in a declarative way. A programmable kernel would allow to fine-tune features of interest and to combine symbolic with sub-symbolic information. Further, collective classification formulations are necessary to be able to classify multiple objects in the image simultaneously based on their features and their graph structure. It is one of the characterizing features of SRL and a form of structured output learning. Finally, the availability of a toolbox implementing qualitative spatial/temporal reasoning concepts would allow an easy and fast integration of qualitative spatial relations and thus, also fast experimentation in vision recognition applications.

9.3 Future Work

Following the remarks made, most of the work could be extended with improved background (e.g., spatial) knowledge that captures better and more complex relations between points of interest, to further increase the performance. This aspect is problem dependent and can be done only with a deep understanding of the domain and application. For instance, the spatial theory defined and implemented in Chapter 4 for the houses domain includes relations such as "touch taller to the right". While this has worked remarkably well in practice, more meaningful and discriminative relations could be defined to capture more information and complex dependencies. However, defining and implementing such relations in practice is time consuming and not trivial. This would be more effectively and efficiently achieved by using available toolboxes that encapsulate principled spatial and temporal theory concepts (e.g., [Cohn and Renz, 2001]).

Then, one could reason in a qualitative way about a diversity of spatial relations, allowing convenient and fast experimentation. Another possibility is to learn from data discriminative relations in a principled way.

Another direction is to extend the applications to new or more complex domains and the datasets used in experiments to more diverse object categories, higher intra-category variation and richer configurations. For instance, the houses dataset can be enhanced with images containing larger buildings having more rows of windows. Similarly, one could directly extend the experiments on generic object recognition (Chapter 5) or grasping capabilities (Chapter 6) with more object categories. Richer protocols involving sequences of interpretations instead of sequences of atoms are another natural direction for the dynamic settings considered in Chapter 8. Although most of the work in this thesis addressed static settings, temporal setups can also gain from SRL. Potential new and interesting applications can be inspired from older work (e.g., understanding aerial images of airports [Russ et al., 1998]) and (re)investigated using current enhanced SRL solutions.

The approaches and results presented in this thesis rely on the traditional task of learning a single predicate or relation independently. However, predicting simultaneously targets based on their features and dependencies can improve performance. For example, in the houses application, the fact that a hypothesis is a door with high probability eliminates the possibility that another touching above hypothesis is also a door at the same time. Thus, another interesting direction is to investigate a collective classification setting for our visual recognition tasks. In collective classification the i.i.d. assumption made by the traditional approach to learning a single predicate or relation no longer holds. A straightforward idea to achieve this is using a local iterative scheme. The prediction process is based on iterative convergence and updates the label of each instance and the labels of the related instances. It continues to do so until the assignments to the labels stabilize. An alternative approach is to define a global objective function to be optimized on the entire relational dataset.

General visual recognition requires flexible matching or interpretation possibilities on many characteristics or features (e.g., there are object categories, such as trees, for which appearance features may be more relevant than high-level information), ability to combine them, skill to use high-level knowledge about the domain, exception handling, and possibility to integrate contextual information. Therefore, visual recognition should have a flexible and multi-representation scheme that covers different aspects and requires a well-organized control strategy of the information. This implies a general and integrated framework rather than bits and pieces of recognition routines. Such a visual recognition system must integrate several aspects: different semantic levels and forms of representation (i.e., sub-symbolic and symbolic), several low and

mid-level visual cues, contextual information, domain and world knowledge, evidence from different modalities. Continuing the idea of general purpose visual recognition, a concrete future step is a declarative kernel language meant to overcome the limitations of kLog. Using this language, the user should be able to program in more detail the definition of a decomposition part by relaxing the notion of a pair of balls.

Another concrete idea is to investigate other SRL systems (e.g., Problog) for similar visual recognition problems in order to evaluate how they perform and thus, establish what SRL language is most convenient and when.

Appendix

Appendix A

Simulated UNO datasets

The artificial datasets were collected by letting two computer agents play Uno. For the positive examples agents play according to the rules, while for the negative examples agents are always cheating with a probability $P = 0.5$ (50% of the time the agent randomly plays one from the cards in their hand). The environment is controlled by an arbiter which deals cards and determines the winner. However, the arbiter's task is solely to manage the card dealing, shuffling and recognizing the end of the game. The arbiter does not check whether the player follow the rules of Uno. Only afterwards the game log is checked to identify whether or not there were illegal moves in the game.

The simulator contains a move function which is Do_Move for the honest agent, whereas a fraudulent agent calls Cheater_Do_Move, as described in Algorithm 1. In some cases he will play a legal card or even pass, but there is also a chance that he discards a random card which might not be a legal move. The game logic build into the arbiter is defined in the function Run_One_Game.

The simulator outputs game protocols in Prolog notation as shown below

```
top_card(1,1,red,5).        card_played(1,1,red,1).        label(1,1,illegal).
top_card(2,1,red,1).        card_played(2,1,yellow,3).     label(2,1,illegal).
top_card(3,1,yellow,3).     joker_played(3,1,wild).        label(3,1,illegal).
...                         ...                            ...
```

Using this simulator, we generated 50 sequences of which 25 contain illegal moves and 25 which contain solely legal moves. The games were simulated using 4 players which results in a sequence length mean of $l_\mu = 27.5$ moves

Algorithm 1 Code used for simulating UNO games between various agents, which can either play according to the rules, or cheat. The arbiter notices fraudulent moves, but does not stop the simulation when they are played.

```
1: function Do_Move(topcard)
2:     if card in hand matches topcard then
3:         return discard randomly a correct card
4:     end if
5:     take card from deck
6:     return null
7: end function
8: function Cheater_Do_Move(topcard)
9:     if Flipping a biased coin yields heads then
10:        return Do_Move(topcard)
11:    else
12:        return discard randomly a card
13:    end if
14: end function
15: function Run_One_Game(agent[])
16:    Shuffle cards
17:    topcard← take first card from deck
18:    deal 7 cards to every agent
19:    p ← random(0, size(agent))                ▷ decide randomly who starts
       (0, 1, 2, . . . , size(agent) − 1)
20:    while Every agent has at least one card do
21:        topcard←agent[p].do_move(topcard)
22:        p ← (p + 1)  mod size(agent)          ▷ Next player's turn
23:    end while
24:    report winner
25:    report illegal_or_not
26: end function
```

per sequence. The sequences consist of a number of logical atoms as shown in the example above. We considered datasets with four levels of artificial noise: $UNO_0 = 0\%$,$UNO_{15} = 15\%$, $UNO_{30} = 30\%$ and $UNO_{50} = 50\%$. A noise level of 30% means that instead of the correct symbol on the card an 'unknown' tag is observed with the following chances: we randomly generate a number between 0 and 1, which if lower than 0.3, one of the correct or 'unknown' symbols is randomly chosen, otherwise the symbol is observed correctly.

Bibliography

[Abdo et al., 2012] Abdo, N., Kretzschmar, H., and Stachniss, C. (2012). From low-level trajectory demonstrations to symbolic actions for planning. In *ICAPS Workshop on Combining Task and Motion Planning for Real-World Applications*.

[Aleotti and Caselli, 2011] Aleotti, J. and Caselli, S. (2011). Part-based robot grasp planning from human demonstration. In *IEEE International Conference on Robotics and Automation*, pages 4554–4560.

[Antanas et al., 2012a] Antanas, L., Frasconi, P., Costa, F., Tuytelaars, T., and De Raedt, L. (2012a). A relational kernel-based framework for hierarchical image understanding. In Gimel'farb, G. L., Hancock, E. R., Imiya, A. I., Kuijper, A., Kudo, M., Shinichiro Omachi, S., Windeatt, T., and Yamada, K., editors, *Lecture Notes in Computer Science*, pages 171–180. Springer.

[Antanas et al., 2013a] Antanas, L., Hoffmann, M., Frasconi, P., Tuytelaars, T., and De Raedt, L. (2013a). A relational kernel-based approach to scene classification. In *Workshop on Applications of Computer Vision*, pages 1–7, Clearwater Beach, Florida.

[Antanas et al., 2013b] Antanas, L., Moreno, P., Figueiredo, R., Neumann, M., Kersting, K., Santos-Victor, J., and De Raedt, L. (2013b). High-level reasoning and low-level learning for grasping: A probabilistic logic pipeline. *IEEE Transactions on Robotics*. Submitted, currently under review.

[Antanas et al., 2014] Antanas, L., van Otterlo, M., Oramas M., J. A., Tuytelaars, T., and De Raedt, L. (2014). There are plenty of places like home: Using relational representations in hierarchies for distance-based image understanding. *Neurocomputing*.

[Antanas et al., 2012b] Antanas, L., van Otterlo, M., Oramas Mogrovejo, J. A., Tuytelaars, T., and De Raedt, L. (2012b). A relational distance-based

framework for hierarchical image understanding. In Latorre Carmona, P., Salvador S'anchez, J., and Fred, A., editors, *International Conference on Pattern Recognition - Applications and Methods*, pages 206–218. Best Paper Award.

[Antanas et al., 2010a] Antanas, L.-A., Gutmann, B., Thon, I., Kersting, K., and De Raedt, L. (2010a). Combining video and sequential statistical relational techniques to monitor card games. In *ICML Workshop on Machine Learning and Games,*, pages 1–6.

[Antanas et al., 2010b] Antanas, L.-A., Gutmann, B., Thon, I., Kersting, K., and De Raedt, L. (2010b). Combining video and sequential statistical relational techniques to monitor card games. In *Annual Belgian-Dutch Conference on Machine Learning,*, pages 1–6.

[Antanas et al., 2009] Antanas, L.-A., Thon, I., van Otterlo, M., Landwehr, N., and De Raedt, L. (2009). Probabilistic logical sequence learning for video. In *Inductive Logic Programming, Leuven, Belgium, 2-4 July 2009*.

[Baltzakis, 2005] Baltzakis, H. (2005). Orca simulator. `http://www.ics.forth.gr/cvrl/_software/orca_setup.exe`.

[Bar-Hillel and Weinshall, 2008] Bar-Hillel, A. and Weinshall, D. (2008). Efficient learning of relational object class models. *International Journal on Computer Vision*, 77(1-3):175–198.

[Barber, 2011] Barber, D. (2011). *Bayesian Reasoning and Machine Learning*. Cambridge University Press.

[Barbu et al., 2010] Barbu, A., Narayanaswamy, S., and Siskind, J. M. (2010). Learning physically-instantiated game play through visual observation. In *IEEE International Conference on Robotics and Automation*, pages 1879–1886. IEEE.

[Barck-Holst et al., 2009] Barck-Holst, C., Ralph, M., Holmar, F., and Kragic, D. (2009). Learning grasping affordance using probabilistic and ontological approaches. In *International Conference on Advanced Robotics*, pages 1–6.

[Barrow and Popplestone, 1971] Barrow, H. G. and Popplestone, R. J. (1971). Relational descriptions in picture processing. *Machine Intelligence*, 6:377–396.

[Bay et al., 2006] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In Leonardis, A., Bischof, H., and Pinz, A., editors, *Computer Vision – European Conference on Computer Vision*, volume 3951 of *Lecture Notes in Computer Science*, pages 404–417. Springer Berlin Heidelberg.

[Belongie et al., 2002] Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522.

[Bennett and Magee, 2007] Bennett, A. and Magee, D. R. (2007). Learning sets of sub-models for spatio-temporal prediction. In *SGAI Conference*, pages 123–136.

[Berardi et al., 2004] Berardi, M., Lapi, M., and Malerba, D. (2004). An integrated approach for automatic semantic structure extraction in document images. In *Document Analysis Systems, volume 3163 of Lecture Notes in Computer Science*, pages 179–190. Springer.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Bishop, 2007] Bishop, C. M. (2007). *Pattern Recognition and Machine Learning*. Springer.

[Bohg et al., 2011] Bohg, J., Johnson-Roberson, M., León, B., Felip, J., Gratal, X., Bergström, N., Kragic, D., and Morales, A. (2011). Mind the gap - robotic grasping under incomplete observation. In *IEEE International Conference on Robotics and Automation*, pages 686–693.

[Bohg and Kragic, 2010] Bohg, J. and Kragic, D. (2010). Learning grasping points with shape context. *Robotics and Autonomous Systems*, 58(4):362–377.

[Bohg et al., 2012] Bohg, J., Welke, K., Leon, B., Do, M., Song, D., Wohlkinger, W., Aldoma, A., Madry, M., Przybylski, M., Asfour, T., Marti, H., Kragic, D., Morales, A., and Vincze, M. (2012). Task-based grasp adaptation on a humanoid robot. In *IFAC Symposium on Robot Control*, pages 779–786.

[Bohlken and Neumann, 2009] Bohlken, W. and Neumann, B. (2009). Generation of rules from ontologies for high-level scene interpretation. In *RuleML*, pages 93–107.

[Bosch et al., 2007] Bosch, A., Zisserman, A., and Munoz, X. (2007). Representing shape with a spatial pyramid kernel. In *ACM International Conference on Image and Video Retrieval*, pages 401–408, New York, NY, USA. ACM.

[Boser and et al., 1992] Boser, B. E. and et al. (1992). A training algorithm for optimal margin classifiers. In *Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press.

[Bouchard, 2005] Bouchard, G. (2005). Hierarchical part-based visual object categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 710–715.

[Boureau et al., 2010] Boureau, Y.-L., Bach, F., LeCun, Y., and Ponce, J. (2010). Learning mid-level features for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2559–2566.

[Boutell et al., 2007] Boutell, M. R., Luo, J., and Brown, C. M. (2007). Scene parsing using region-based generative models. *IEEE Transactions on Multimedia*, 9(1):136–146.

[Brooks, 1981] Brooks, R. A. (1981). Symbolic reasoning among 3D models and 2D images. *AI Journal*, 17:285–348.

[Bunke and Sanfeliu, 1990] Bunke, H. and Sanfeliu, A. (1990). *Syntactic and Structural Pattern Recognition: Theory and Applications*. World Scientific Pub Co Inc.

[Bunke and Shearer, 1998] Bunke, H. and Shearer, K. (1998). A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19(3-4):255–259.

[Busygin, 2006] Busygin, S. (2006). A new trust region technique for the maximum weight clique problem. *Discrete Applied Mathematics*, 154(15):2080–2096.

[Caetano et al., 2009] Caetano, T. S., McAuley, J. J., Cheng, L., Le, Q. V., and Smola, A. J. (2009). Learning graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1048–1058.

[Chang and Lin, 2011] Chang, C. and Lin, C. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27.

[Choi et al., 2012] Choi, M. J., Torralba, A., and Willsky, A. S. (2012). A tree-based context model for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):240–252.

[Clowes, 1971] Clowes, M. B. (1971). On seeing things. *Artificial intelligence*, 2(1):79–116.

[Cocora et al., 2006] Cocora, A., Kersting, K., Plagemann, C., Burgard, W., and De Raedt, L. (2006). Learning relational navigation policies. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2792–2797.

[Cohn et al., 2008] Cohn, A. G., Hogg, D. C., Möller, R., and Neumann, B., editors (2008). *Logic and Probability for Scene Interpretation*. Dagstuhl Seminar Proceedings.

[Cohn and Renz, 2001] Cohn, A. G. and Renz, J. (2001). Qualitative spatial representation and reasoning: An overview. *Fundamenta Informaticae*, 46:1–2.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20:273–297. 10.1007/BF00994018.

[Costa and De Grave, 2010] Costa, F. and De Grave, K. (2010). Fast neighborhood subgraph pairwise distance kernel. In *International Conference on Machine Learning*, pages 255–262.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893.

[Damen and Hogg, 2009] Damen, D. and Hogg, D. (2009). Recognizing linked events: Searching the space of feasible explanations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 927–934.

[Dang and Allen, 2012] Dang, H. and Allen, P. K. (2012). Semantic grasping: Planning robotic grasps functionally suitable for an object manipulation task. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1311–1317.

[Das, 1992] Das, S. K. (1992). *Deductive Databases and Logic Programming*. Addison-Wesley.

[Datar and Indyk, 2004] Datar, M. and Indyk, P. (2004). Locality-sensitive hashing scheme based on p-stable distributions. In *Annual Symposium on Computational Geometry*, pages 253–262.

[De Raedt, 2008] De Raedt, L. (2008). *Logical and Relational Learning*. Cognitive Technologies. Springer.

[De Raedt and Ramon, 2009] De Raedt, L. and Ramon, J. (2009). Deriving distance metrics from generality relations. *Pattern Recognition Letters*, 30(3):187–191.

[De Raedt and Thon, 2010] De Raedt, L. and Thon, I. (2010). Probabilistic rule learning. In *Inductive Logic Programming*, pages 47–58.

[Desai et al., 2009] Desai, C., Ramanan, D., and Fowlkes, C. (2009). Discriminative models for multi-class object layout. In *International Conference of Computer Vision*, pages 229–236. Ieee.

[Deselaers and Ferrari, 2010] Deselaers, T. and Ferrari, V. (2010). Global and efficient self-similarity for object classification and detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1633–1640.

[D'Este and Sammut, 2008] D'Este, C. and Sammut, C. (2008). Learning and generalising semantic knowledge from object scenes. *Robot. Auton. Syst.*, 56(11):891–900.

[Detry et al., 2012a] Detry, R., Ek, C. H., Madry, M., and Kragic, D. (2012a). Compressing grasping experience into a dictionary of prototypical grasp-predicting parts. In *International Workshop on Human-Friendly Robotics*.

[Detry et al., 2013] Detry, R., Ek, C. H., Madry, M., and Kragic, D. (2013). Learning a dictionary of prototypical grasp-predicting parts from grasping experience. In *IEEE International Conference on Robotics and Automation*. To appear.

[Detry et al., 2012b] Detry, R., Ek, C. H., Madry, M., Piater, J., and Kragic, D. (2012b). Generalizing grasps across partly similar objects. In *IEEE International Conference on Robotics and Automation*.

[Dickinson and Davis, 1987] Dickinson, S. and Davis, L. (1987). *An Expert Vision System for Autonomous Land Vehicle Road Following*. Maryland University, Center for Automation Research.

[Dietterich et al., 2004] Dietterich, T., Ashenfelter, A., and Bulatov, Y. (2004). Training conditional random fields via gradient tree boosting. In *Proc. 21st International Conf. on Machine Learning*, pages 217–224. ACM.

[Divvala et al., 2009] Divvala, S., Hoiem, D., Hays, J., Efros, A., and Hebert, M. (2009). An empirical study of context in object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1271–1278.

[Draper et al., 1989] Draper, B. A., Collins, R. T., Brolio, J., Hanson, A. R., and Riseman, E. M. (1989). The schema system. *International Journal of Computer Vision*, 2(3):209–250.

[Draper et al., 1996] Draper, B. A., Hanson, A. R., and Riseman, E. M. (1996). Knowledge-directed vision: Control, learning, and integration.

[Dubba et al., 2010] Dubba, K. S. R., Cohn, A. G., and Hogg, D. C. (2010). Event model learning from complex videos using inductive logic programming. In *European Conference on Artificial Intelligence*, pages 93–98.

[Duchenne et al., 2011] Duchenne, O., Joulin, A., and Ponce, J. (2011). A graph-matching kernel for object categorization. In *IEEE International Conference on Computer Vision*, pages 1792–1799.

[El-Khoury and Sahbani, 2010] El-Khoury, S. and Sahbani, A. (2010). A new strategy combining empirical and analytical approaches for grasping unknown 3D objects. *Robot. Auton. Syst.*, 58(5):497–507.

[Epshtein and Ullman, 2007] Epshtein, B. and Ullman, S. (2007). Semantic hierarchies for recognizing objects and parts. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[Erkan et al., 2010] Erkan, A., Kroemer, O., Detry, R., Altun, Y., Piater, J., and Peters, J. (2010). Learning probabilistic discriminative models of grasp affordances under limited supervision. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1586–1591.

[Esposito et al., 2001] Esposito, F., Malerba, D., and Marengo, V. (2001). Inductive learning from numerical and symbolic data: An integrated framework. *Intell. Data Anal.*, 5(6):445–461.

[Esposito et al., 1992] Esposito, F., Malerba, D., and Semeraro, G. (1992). Classification in noisy environments using a distance measure between structural symbolic descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:390–402.

[Everingham et al., 2008] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2008). The PASCAL Visual Object Classes Challenge 2008. `http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html`.

[Everingham et al., 2012] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2012). The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[Fan et al., 2008] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. *J. Machine Learning Res.*, 9:1871–1874.

[Farid and Sammut, 2012] Farid, R. and Sammut, C. (2012). Plane-based object categorization using relational learning. In *International Conference on Inductive Logic Programming - Machine Learning Journal*, page To appear.

[Fei-Fei, 2013] Fei-Fei, L. (2013). Computer vision: A quest for visual intelligence. BigThinkers. Lecture conducted from Yahoo Labs.

[Fei-Fei and Perona, 2005] Fei-Fei, L. and Perona, P. (2005). A Bayesian hierarchical model for learning natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 524–531.

[Felzenszwalb et al., 2010] Felzenszwalb, P., Girshick, R., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627 –1645.

[Felzenszwalb and Huttenlocher, 2005] Felzenszwalb, P. and Huttenlocher, D. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79.

[Fergus et al., 2007] Fergus, R., Perona, P., and Zisserman, A. (2007). Weakly supervised scale-invariant learning of models for visual recognition. *International Journal of Computer Vision*, 71(3):273–303.

[Ferilli et al., 2003] Ferilli, S., Mauro, N. D., Basile, T. M. A., and Esposito, F. (2003). A complete subsumption algorithm. In *AI*IA 2003*, pages 23–26.

[Fern, 2005] Fern, A. (2005). A simple-transition model for relational sequences. In *International Joint Conference on Artificial Intelligence*, pages 696–701.

[Ferrari et al., 2008] Ferrari, V., Fevrier, L., Jurie, F., and Schmid, C. (2008). Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1):36–51.

[Ferraté et al., 1988] Ferraté, G., Pavlidis, T., Sanfeliu, A., and Bunke, H. (1988). *Syntactic and Structural Pattern Recognition*. NATO ASI Series / Computer and Systems Sciences. Springer-Verlag.

[Fidler et al., 2009] Fidler, S., Boben, M., and Leonardis, A. (2009). Learning Hierarchical Compositional Representations of Object Structure. In *Object Categorization: Computer and Human Vision Perspectives*. Cambridge University Press.

[Fidler and Leonardis, 2007] Fidler, S. and Leonardis, A. (2007). Towards scalable representations of object categories: Learning a hierarchy of parts. In *Conference on Computer Vision and Pattern Recognition*.

[Fidler et al., 2013] Fidler, S., Mottaghi, R., Yuille, A. L., and Urtasun, R. (2013). Bottom-up segmentation for top-down detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3294–3301. IEEE.

[Fierens et al., 2011] Fierens, D., Van den Broeck, G., Thon, I., Gutmann, B., and De Raedt, L. (2011). Inference in probabilistic logic programs using weighted CNF's. In Gagliardi Cozman, F. and Pfeffer, A., editors, *Conference on Uncertainty in Artificial Intelligence*, pages 211–220.

[Fischinger et al., 2013] Fischinger, D., Vincze, M., and Jiang, Y. (2013). Learning grasps for unknown objects in cluttered scenes. In *IEEE International Conference on Robotics and Automation*.

[Fischler and Elschlager, 1973] Fischler, M. and Elschlager, R. (1973). The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92.

[Flach, 2012] Flach, P. (2012). *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. Cambridge University Press.

[Frasconi et al., 2012] Frasconi, P., Costa, F., De Raedt, L., and De Grave, K. (2012). klog: A language for logical and relational learning with kernels. *CoRR*, abs/1205.3981.

[Fu, 1974] Fu, K. (1974). *Syntactic Methods in Pattern Recognition*. Elsevier Science.

[Fu, 1982] Fu, K. (1982). *Syntactic Pattern Recognition and Applications*. Prentice-Hall advanced reference series: Computer science. Prentice-Hall.

[Fu, 1983] Fu, K. (1983). A syntactic-semantic approach to pictorial pattern analysis. In *PDA83*, pages 133–146.

[Galleguillos and Belongie, 2010] Galleguillos, C. and Belongie, S. (2010). Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, 114(6):712–722.

[Galleguillos et al., 2008] Galleguillos, C., Rabinovich, A., and Belongie, S. (2008). Object categorization using co-occurrence, location and appearance. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society.

[Garcia et al., 2012] Garcia, S., Derrac, J., Cano, J. R., and Herrera, F. (2012). Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):417–435.

[Garcia-Molina et al., 2008] Garcia-Molina, H., Ullman, J. D., and Widom, J. (2008). *Database Systems: The Complete Book*. Prentice Hall Press, Upper Saddle River, NJ, USA, 2 edition.

[Gartner, 2008] Gartner, T. (2008). *Kernels for structured data*, volume 72. World Scientific.

[Gärtner et al., 2004] Gärtner, T., Lloyd, J. W., and Flach, P. A. (2004). Kernels and distances for structured data. *Machine Learning*, 57(3):205–232.

[Getoor et al., 2000] Getoor, L., Koller, D., Taskar, B., and Friedman, N. (2000). Learning probabilistic relational models with structural uncertainty. In *ICML Workshop on Attribute-Value and Relational Learning:Crossing the Boundaries*, pages 13–20.

[Getoor and Taskar, 2007] Getoor, L. and Taskar, B. (2007). *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

[Girshick et al., 2011] Girshick, R. B., Felzenszwalb, P. F., and McAllester, D. A. (2011). Object detection with grammar models. In *Advances in Neural Information Processing Systems*, pages 442–450.

[Goad, 1983] Goad, C. (1983). *Special Purpose Automatic Programming for 3D Model-Based Vision*. Defense Technical Information Center.

[González and Thomason, 1978] González, R. and Thomason, M. (1978). *Syntactic Pattern Recognition: An Introduction*. Applied mathematics and computation. Addison-Wesley Pub. Co., Advanced Book Program.

[Grauman and Darrell, 2005] Grauman, K. and Darrell, T. (2005). The pyramid match kernel: Discriminative classification with sets of image features. In *International Conference on Computer Vision*, pages 1458–1465.

[Gries et al., 2010] Gries, O., Möller, R., Nafissi, A., Rosenfeld, M., Sokolski, K., and Wessel, M. (2010). A probabilistic abduction engine for media interpretation based on ontologies. In *Proceedings of the Fourth International Conference on Web Reasoning and Rule Systems*, pages 182–194, Berlin, Heidelberg. Springer-Verlag.

[Grimson and Lozano-Pérez, 1987] Grimson, W. and Lozano-Pérez, T. (1987). Localizing overlapping parts by searching the interpretation tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):469–482.

[Gupta and Davis, 2008] Gupta, A. and Davis, L. (2008). Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. *Computer Vision–European Conference on Computer Vision*, pages 16–29.

[Guthrie et al., 2006] Guthrie, D., Allison, B., Liu, W., Guthrie, L., and Wilks, Y. (2006). A closer look at skip-gram modelling. In *International Conference on Language Resources and Evaluation*, pages 1222–1225, Genoa, Italy.

[Gutmann and Kersting, 2006] Gutmann, B. and Kersting, K. (2006). TildeCRF: Conditional random fields for logical sequences. In *European Conference on Machine Learning*, pages 174–185.

[Guzmán, 1968] Guzmán, A. (1968). Decomposition of a visual scene into three-dimensional bodies. In *Proceedings of the Fall Joint Computing Conference, part I*, AFIPS '68 (Fall, part I), pages 291–304, New York, NY, USA. ACM.

[Guzmán, 1971] Guzmán, A. (1971). Analysis of curved line drawings using context and global information. In Meltzer, B. and Mitchie, D., editors, *Machine Intelligence (6)*, pages 325–376. Edinburgh University Press.

[Han and Zhu, 2009] Han, F. and Zhu, S.-C. (2009). Bottom-up/top-down image parsing with attribute grammar. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):59–73.

[Hanheide et al., 2011] Hanheide, M., Gretton, C., Dearden, R., Hawes, N., Wyatt, J. L., Pronobis, A., Aydemir, A., Göbelbecker, M., and Zender, H. (2011). Exploiting probabilistic knowledge under uncertain sensing for efficient robot behaviour. In *International Joint Conference on Artificial Intelligence*, pages 2442–2449.

[Hanson and Riseman, 1978] Hanson, A. R. and Riseman, E. M. (1978). VISIONS: A computer system for interpreting scenes. In Hanson, A. R. and Riseman, E. M., editors, *Computer Vision Systems*. Academic Press, New York.

[Haralick, 1983] Haralick, R. (1983). *Pictorial Data Analysis*. NATO ASI series: Computer and system sciences. Springer-Verlag.

[Harchaoui and Bach, 2007] Harchaoui, Z. and Bach, F. (2007). Image classification with segmentation graph kernels. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[Hart et al., 2005] Hart, S., Grupen, R. A., and Jensen, D. (2005). A relational representation for procedural task knowledge. In *Conference on Artificial Intelligence*, pages 1280–1285.

[Hartz, 2009] Hartz, J. (2009). Learning probabilistic structure graphs for classification and detection of object structures. In *Machine Learning and Applications*, pages 5–11. IEEE.

[Hartz et al., 2009] Hartz, J., Hotz, L., Neumann, B., and Terzić, K. (2009). Automatic incremental model learning for scene interpretation. In *International Conference on Computational Intelligence*, Honolulu, Hawaii.

[Hartz and Neumann, 2007] Hartz, J. and Neumann, B. (2007). Learning a knowledge base of ontological concepts for high-level scene interpretation. In *Machine Learning and Applications*, pages 436–443. IEEE.

[Haussler, 1999] Haussler, D. (1999). Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, University of California at Santa Cruz.

[Hazarika and Bhowmick, 2012] Hazarika, S. M. and Bhowmick, A. (2012). Learning rules of a card game from video. *Artificial Intelligence Review*, 38(1):55–65.

[Hoiem et al., 2007] Hoiem, D., Efros, A. A., and Hebert, M. (2007). Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172.

[Horváth et al., 2001] Horváth, T., Wrobel, S., and Bohnebeck, U. (2001). Relational instance-based learning with lists and terms. *Machine Learning*, 43(1/2):53–80.

[Hotz and Neumann, 2010] Hotz, L. and Neumann, B. (2010). Learning and recognizing structures in façade scenes (eTRIMS) - A retrospective. *KI*, 24(1):63–68.

[Huffman, 1971] Huffman, D. A. (1971). Impossible Objects as Nonsense Sentences. *Machine Intelligence*, 6:295–323.

[Hummel, 2010] Hummel, B. (2010). *Description Logic for Scene Understanding*. Suedwestdeutscher Verlag fuer Hochschulschriften, Germany.

[Ikeuchi, 1987] Ikeuchi, K. (1987). Precompiling a geometrical model into an interpretation tree for object recognition in bin-picking tasks. In *Proc. of the Image Understanding Workshop*, pages 321–339, Los Angeles, CA.

[Jia et al., 2012] Jia, Y., Huang, C., and Darrell, T. (2012). Beyond spatial pyramids: Receptive field learning for pooled image features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3370–3377.

[Jiang et al., 2011] Jiang, Y., Moseson, S., and Saxena, A. (2011). Efficient grasping from rgbd images: Learning using a new rectangle representation. In *IEEE International Conference on Robotics and Automation*, pages 3304–3311.

[Kaiser, 2012] Kaiser, Ł. (2012). Learning games from videos guided by descriptive complexity. In *Conference on Artificial Intelligence*, pages 963–970. AAAI Press.

[Kanade, 1977] Kanade, T. (1977). Model representations and control structures in image understanding. In *International Joint Conference on Artificial Intelligence - Volume 2*, pages 1074–1082, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Kappler et al., 2010] Kappler, D., Chang, L. Y., Przybylski, M., Pollard, N., Asfour, T., and Dillmann, R. (2010). Representation of pre-grasp strategies for object manipulation. In *IEEE/RAS International Conference on Humanoid Robots*.

[Karlinsky et al., 2010] Karlinsky, L., Dinerstein, M., Harari, D., and Ullman, S. (2010). The chains model for detecting parts by their context. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 25–32.

[Karwath and Kersting, 2007] Karwath, A. and Kersting, K. (2007). Relational sequence alignments and logos. In Muggleton, S., Otero, R., and Tamaddoni-Nezhad, A., editors, *Inductive Logic Programming*, pages 290–304. Springer-Verlag, Berlin, Heidelberg.

[Kato et al., 2000] Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K., and Tachibana, K. (2000). Virtual object manipulation on a table-top AR environment. In *International Symposium on Augmented Reality*, page 111.

[Kersting et al., 2008] Kersting, K., De Raedt, L., Gutmann, B., Karwath, A., and Landwehr, N. (2008). Relational sequence learning. In *Probabilistic Inductive Logic Programming*, volume 4911 of *Lecture Notes in Computer Science*, pages 28–55. Springer.

[Kersting et al., 2006] Kersting, K., De Raedt, L., and Raiko, T. (2006). Logical hidden Markov models. *Journal of Artificial Intelligence Research*, 25:2006.

[Keselman and Dickinson, 2005] Keselman, Y. and Dickinson, S. (2005). Generic model abstraction from examples. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1141–1156.

[Khoshafian and Copeland, 1986] Khoshafian, S. and Copeland, G. (1986). Object identity. In *ACM Conference on Object-Oriented Programming Systems, Languages, and Applications*, pages 406–416.

[Kimmig and Costa, 2012] Kimmig, A. and Costa, F. (2012). Link and node prediction in metabolic networks with probabilistic logic. In Berthold, M. R., editor, *Lecture Notes in Artificial Intelligence*. Springer.

[Kirsten et al., 2000] Kirsten, M., Wrobel, S., and Horváth, T. (2000). Distance based approaches to relational learning and clustering. *Relational Data Mining*, pages 213–230.

[Kjellström et al., 2008] Kjellström, H., Romero, J., and Kragic, D. (2008). Visual recognition of grasps for human-to-robot mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3192–3199.

[Koutsourakis et al., 2009] Koutsourakis, P., Simon, L., Teboul, O., Tziritas, G., and Paragios, N. (2009). Single view reconstruction using shape grammars for urban environments. In *International Conference on Computer Vision*, pages 1795–1802.

[Kraft et al., 2010] Kraft, D., Detry, R., Pugeault, N., Baseski, E., Guerin, F., Piater, J. H., and Krüger, N. (2010). Development of object and grasping knowledge by robot exploration. *IEEE T. Autonomous Mental Development*, 2(4):368–383.

[Kraft et al., 2009] Kraft, D., Detry, R., Pugeault, N., Baseski, E., Piater, J. H., and Krüger, N. (2009). Learning objects and grasp affordances through autonomous exploration. In *International Conference on Computer Vision Systems*, pages 235–244.

[Kreutzmann et al., 2009] Kreutzmann, A., Terzić, K., and Neumann, B. (2009). Context-aware classification for incremental scene interpretation. In *Workshop on Use of Context in Vision Processing*, pages 1–6, New York, NY, USA. ACM.

[Kulick et al., 2013] Kulick, J., Toussaint, M., Lang, T., and Lopes, M. (2013). Active learning for teaching a robot grounded relational symbols. In *International Joint Conferences on Artificial Intelligence*.

[Körtgen et al., 2003] Körtgen, M., Novotni, M., and Klein, R. (2003). 3D shape matching with 3D shape contexts. In *The 7th Central European Seminar on Computer Graphics*.

[Ladický et al., 2010] Ladický, L., Sturgess, P., Alahari, K., Russell, C., and Torr, P. H. S. (2010). What, where and how many? Combining object detectors and CRFs. In *European Conference on Computer Vision: Part IV*, European Conference on Computer Vision, pages 424–437, Berlin, Heidelberg. Springer-Verlag.

[Lafferty et al., 2001] Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, pages 282–289.

[Landwehr, 2009] Landwehr, N. (2009). *Trading Expressivity for Efficiency in Statistical Relational Learning*. PhD thesis, Informatics Section, Department of Computer Science, Faculty of Engineering Science. De Raedt, Luc (supervisor).

[Landwehr and De Raedt, 2007] Landwehr, N. and De Raedt, L. (2007). r-Grams: Relational grams. In *International Joint Conference on Artificial Intelligence*, pages 907–912.

[Lang and Toussaint, 2010] Lang, T. and Toussaint, M. (2010). Planning with noisy probabilistic relational rules. *Journal of Artificial Intelligence Research*, 39:1–49.

[Lazebnik et al., 2006] Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178, Washington, DC, USA. IEEE Computer Society.

[Lee and Grauman, 2012] Lee, Y. J. and Grauman, K. (2012). Object-graphs for context-aware visual category discovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):346–358.

[Lenz et al., 2013] Lenz, I., Lee, H., and Saxena, A. (2013). Deep learning for detecting robotic grasps. *CoRR*, abs/1301.3592.

[Leow and Miikkulainen, 1994] Leow, W. and Miikkulainen, R. (1994). Visor: Schema-based scene analysis with structured neural networks. *Neural Processing Letters*, 1:18–23.

[Li et al., 2012] Li, C., Parikh, D., and Chen, T. (2012). Automatic discovery of groups of objects for scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2735 –2742.

[Li et al., 2009] Li, L.-J., Socher, R., and Fei-Fei, L. (2009). Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[Li-Jia Li and Fei-Fei, 2010] Li-Jia Li, Hao Su, E. P. X. and Fei-Fei, L. (2010). Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *Advances in Neural Information Processing Systems*, Vancouver, Canada.

[Limketkai et al., 2005] Limketkai, B., Liao, L., and Fox, D. (2005). Relational object maps for mobile robots. In *International Joint Conference on Artificial Intelligence*, pages 1471–1476.

[Lippow, 2010] Lippow, M. A. (2010). *Weighted geometric grammars for object detection in context*. PhD thesis, Massachusetts Institute of Technology.

[Lippow et al., 2008] Lippow, M. A., Kaelbling, L. P., and Lozano-Perez, T. (2008). Learning grammatical models for object recognition. In Cohn, A. G., Hogg, D. C., Möller, R., and Neumann, B., editors, *Logic and Probability for Scene Interpretation, Technical Report*, number 08091 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal on Computer Vision*, 60:91–110.

[Lozin and Milanic, 2010] Lozin, V. and Milanic, M. (2010). On the maximum independent set problem in subclasses of planar graphs. *Journal of Graph Algorithms and Applications*, 14:269–286.

[Lutz and Bühlmann, 2006] Lutz, R. and Bühlmann, P. (2006). Conjugate direction boosting. *Journal of Computational and Graphical Statistics*, 15(2):287–311.

[MacGregor and Bates, 1987] MacGregor, R. and Bates, R. (1987). The loom knowledge representation language. Technical report, DTIC Document.

[Madry et al., 2012a] Madry, M., Song, D., Ek, C. H., and Kragic, D. (2012a). "Robot bring me something to drink from": Object representation for transferring task specific grasps. In *ICRA Workshop on Semantic Perception, Mapping and Exploration*.

[Madry et al., 2012b] Madry, M., Song, D., and Kragic, D. (2012b). From object categories to grasp transfer using probabilistic reasoning. In *IEEE International Conference on Robotics and Automation*, pages 1716–1723.

[Malerba, 2003] Malerba, D. (2003). Learning recursive theories in the normal inductive logic programming setting. *Fundamenta Informaticae*, 57(1):39–77.

[Malisiewicz and Efros, 2009] Malisiewicz, T. and Efros, A. A. (2009). Beyond categories: The visual memex model for reasoning about object relationships. In *Advances in Neural Information Processing Systems*.

[Manning and Schütze, 1999] Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

[Martinović and Van Gool, 2013] Martinović, A. and Van Gool, L. (2013). Bayesian grammar learning for inverse procedural modeling. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[Marton et al., 2009] Marton, Z. C., Rusu, R. B., Jain, D., Klank, U., and Beetz, M. (2009). Probabilistic categorization of kitchen objects in table settings with a composite sensor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA.

[Matsuyama and Hwang, 1985] Matsuyama, T. and Hwang, V. S. (1985). Sigma: A framework for image understanding - integration of bottom-up and top-down analysis. In *International Joint Conference on Artificial Intelligence*, pages 908–915.

[McAuley et al., 2009] McAuley, J. J., de Campos, T., Csurka, G., and Perronnin, F. (2009). Hierarchical image-region labeling via structured learning. In *British Machine Vision Conference*.

[McKeown et al., 1985] McKeown, D. M., Harvey, W. A., and McDermott, J. (1985). Rule-based interpretation of aerial imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(5):570–585.

[Meert et al., 2008] Meert, W., Struyf, J., and Blockeel, H. (2008). Learning ground CP-logic theories by leveraging Bayesian network learning techniques. *Fundam. Inform.*, 89(1):131–160.

[Menchetti et al., 2005] Menchetti, S., Costa, F., and Frasconi, P. (2005). Weighted decomposition kernels. In *International Conference on Machine learning*, pages 585–592. ACM.

[Michalski et al., 1986] Michalski, S. R., Carbonell, G. J., and Mitchell, M. T., editors (1986). *Machine Learning: An Artificial Intelligence Approach, Volume II*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Mikolajczyk and Schmid, 2005] Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630.

[Moayer and Fu, 1976] Moayer, B. and Fu, K.-S. (1976). A tree system approach for fingerprint pattern recognition. *IEEE Transactions on Computers*, 25(3):262–274.

[Moldovan et al., 2013a] Moldovan, B., Antanas, L., and Hoffmann, M. (2013a). Opening doors: An initial SRL approach. In *Lecture Notes in Computer Science*.

[Moldovan et al., 2013b] Moldovan, B., Moreno, P., and van Otterlo, M. (2013b). On the use of probabilistic relational affordance models for sequential manipulation tasks in robotics. In *IEEE International Conference on Robotics and Automation*.

[Moldovan et al., 2012] Moldovan, B., Moreno, P., van Otterlo, M., Santos-Victor, J., and De Raedt, L. (2012). Learning relational affordance models for robots in multi-object manipulation tasks. In *IEEE International Conference on Robotics and Automation*, pages 4373 –4378.

[Montesano and Lopes, 2009] Montesano, L. and Lopes, M. (2009). Learning grasping affordances from local visual descriptors. In *IEEE International Conference on Development and Learning*, pages 1–6, Washington, DC, USA. IEEE Computer Society.

[Montesano and Lopes, 2012] Montesano, L. and Lopes, M. (2012). Active learning of visual descriptors for grasping using non-parametric smoothed beta distributions. *IEEE/RAS International Conference on Humanoid Robots*, 60(3):452–462.

[Moreno et al., 2011] Moreno, P., Hornstein, J., and Santos-Victor, J. (2011). Learning to grasp from point clouds. Technical Report Vislab-TR001/2011, Department of Electrical and Computers Engineering, Instituto Superior Técnico, Portugal.

[Muggleton and Buntine, 1988] Muggleton, S. and Buntine, W. L. (1988). Machine invention of first-order predicates by inverting resolution. In *Machine Learning*, pages 339–352.

[Muja and Ciocarlie, 2012] Muja, M. and Ciocarlie, M. (2012). Table top segmentation package. `http://www.ros.org/wiki/tabletop_object_detector`.

[Müller et al., 2007] Müller, P., Zeng, G., Wonka, P., and Van Gool, L. (2007). Image-based procedural modeling of facades. *ACM Transactions on Graphics*, 26(3):85.

[Mundy, 2006] Mundy, J. L. (2006). Object recognition in the geometric era: A retrospective. In *Toward CategoryLevel Object Recognition, volume 4170 of Lecture Notes in Computer Science*, pages 3–29. Springer.

[Needham et al., 2008] Needham, C., Santos, P., and Magee, D. (2008). Inductive learning spatial attention. *Control and Automation*, 19:316–326.

[Needham et al., 2005] Needham, C. J., Santos, P. E., Magee, D. R., Devin, V., Hogg, D. C., and Cohn, A. G. (2005). Protocols from perceptual observations. *Artificial Intelligence*, 167:103–136.

[Neumann and Möller, 2008] Neumann, B. and Möller, R. (2008). On scene interpretation with description logics. *Image Vision Comput.*, 26(1):82–101.

[Neumann et al., 2012a] Neumann, M., Garnett, R., Moreno, P., Patricia, N., and Kersting, K. (2012a). Propagation kernels for partially labeled graphs. In *ICML–2012 Workshop on Mining and Learning with Graphs (MLG–2012)*, Edinburgh, UK.

[Neumann et al., 2013] Neumann, M., Moreno, P., Antanas, L., Garnett, R., and Kersting, K. (2013). Graph kernels for object category prediction in task–dependent robot grasping. In *Eleventh Workshop on Mining and Learning with Graphs*.

[Neumann et al., 2012b] Neumann, M., Patricia, N., Garnett, R., and Kersting, K. (2012b). Efficient graph kernels by randomization. In *ECML/PKDD*, pages 378–393.

[Nevatia and Binford, 1977] Nevatia, R. and Binford, T. O. (1977). Description and recognition of curved objects. *Artificial Intelligence*, 8(1):77–98.

[Nienhuys-Cheng, 1997] Nienhuys-Cheng, S.-H. (1997). Distance between herbrand interpretations: A measure for approximations to a target concept. In *Inductive Logic Programming*, pages 213–226.

[Nilsson, 2009] Nilsson, N. J. (2009). *The Quest for Artificial Intelligence*. Cambridge University Press, New York, NY, USA, 1st edition.

[Nowozin et al., 2010] Nowozin, S., Gehler, P. V., and Lampert, C. H. (2010). On parameter learning in CRF-based approaches to object class image segmentation. In *European Conference on Computer Vision: Part VI*, pages 98–111, Berlin, Heidelberg. Springer-Verlag.

[Nowozin et al., 2011] Nowozin, S., Rother, C., Bagon, S., Sharp, T., Yao, B., and Kohli, P. (2011). Decision tree fields. In *IEEE International Conference on Computer Vision*, pages 1668–1675.

[Ohta, 1985] Ohta, Y. (1985). *Knowledge-based Interpretation of Outdoor Natural Color Scenes*. Pitman Publishing, Inc., Marshfield, MA, USA.

[Ohta et al., 1979] Ohta, Y., Kanade, T., and Sakai, T. (1979). A production system for region analysis. In *International Joint Conference on Artificial Intelligence*, pages 684–686.

[Oliva and Torralba, 2001] Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42:145–175.

[Oliva and Torralba, 2006] Oliva, A. and Torralba, A. (2006). Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research*, 155:23–36.

[Ommer and Buhmann, 2010] Ommer, B. and Buhmann, J. M. (2010). Learning the compositional nature of visual object categories for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:501–516.

[Östergård, 2002] Östergård, P. R. J. (2002). A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, 120:197–207.

[Pandey and Lazebnik, 2011] Pandey, M. and Lazebnik, S. (2011). Scene recognition and weakly supervised object localization with deformable part-based models. In *International Conference on Computer Vision*.

[Parikh and Chen, 2007] Parikh, D. and Chen, T. (2007). Hierarchical semantics of objects (hSOs). *IEEE International Conference on Computer Vision*, 0:1–8.

[Parizi et al., 2012] Parizi, S. N., Oberlin, J. G., and Felzenszwalb, P. F. (2012). Reconfigurable models for scene recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2775 –2782.

[Park et al., 2010] Park, D., Ramanan, D., and Fowlkes, C. (2010). Multiresolution models for object detection. In *European Conference on Computer Vision*, pages 241–254, Berlin, Heidelberg. Springer-Verlag.

[Pavlidis and Ali, 1979] Pavlidis, T. and Ali, F. (1979). Hierarchical Syntactic Shape Analyser. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1:2–9.

[Peraldi et al., 2009] Peraldi, I. S. E., Kaya, A., and Möller, R. (2009). Formalizing multimedia interpretation based on abduction over description logic aboxes. In *Description Logics*.

[Peraldi et al., 2011] Peraldi, I. S. E., Kaya, A., and Möller, R. (2011). Logical formalization of multimedia interpretation. In Paliouras, G., Spyropoulos, C. D., and Tsatsaronis, G., editors, *Knowledge-Driven Multimedia Information Extraction and Ontology Evolution*, volume 6050 of *Lecture Notes in Computer Science*, pages 110–133. Springer.

[Petrou, 2008] Petrou, M. (2008). The tower of knowledge: A novel architecture for organising knowledge combining logic and probability. In *Logic and Probability for Scene Interpretation*, Dagstuhl Seminar Proceedings.

[Pineda et al., 2009] Pineda, G. F., Koga, H., and Watanabe, T. (2009). Unsupervised object discovery from images by mining local features using hashing. In *Iberoamerican Congress on Pattern Recognition*, pages 978–985.

[Pinz et al., 2009] Pinz, A. J., Bischof, H., Kropatsch, W. G., Schweighofer, G., Haxhimusa, Y., Opelt, A., and Ion, A. (2009). Representations for cognitive vision: A review of appearance-based, spatio-temporal, and graph-based approaches. *Electronic Letters on Computer Vision and Image Analysis*, 7(2):35–61.

[Platt, 1999] Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press.

[Ponce and Kriegman, 1990] Ponce, J. and Kriegman, D. J. (1990). Computing exact aspect graphs of curved objects: Parametric surfaces. In Shrobe, H. E.,

Dietterich, T. G., and Swartout, W. R., editors, *Conference on Artificial Intelligence*, pages 1074–1079. AAAI Press / The MIT Press.

[Poole, 1993] Poole, D. (1993). Probabilistic Horn abduction and Bayesian networks. *Artif. Intell.*, 64(1):81–129.

[Popovic et al., 2010] Popovic, M., Kraft, D., Bodenhagen, L., Baseski, E., Pugeault, N., Kragic, D., Asfour, T., and Krüger, N. (2010). A strategy for grasping unknown objects based on co-planarity and colour information. *Robotics and Autonomous Systems*, 58(5):551–565.

[Porway et al., 2007a] Porway, J., Yao, B., and Zhu, S. C. (2007a). Learning compositional models for object categories from small sample sets. Technical report, UCLA.

[Porway et al., 2007b] Porway, J., Yao, Z. Y., and Zhu, S. C. (2007b). Learning an and-or graph for modeling and recognizing object categories. Technical report, UCLA.

[Prats et al., 2007] Prats, M., Sanz, P. J., and Pobil, A. P. D. (2007). Task-oriented grasping using hand preshapes and task frames. In *IEEE International Conference on Robotics and Automation*, pages 1794–1799. IEEE.

[Preparata and Ray, 1972] Preparata, F. P. and Ray, S. R. (1972). An approach to artificial nonsymbolic cognition. *Information Science*, 4(1):65–86.

[Quattoni et al., 2004] Quattoni, A., Collins, M., and Darrell, T. (2004). Conditional random fields for object recognition. In *Advances in Neural Information Processing Systems*, pages 1097–1104. MIT Press.

[Quattoni and Torralba, 2009] Quattoni, A. and Torralba, A. (2009). Recognizing indoor scenes. *IEEE Conference on Computer Vision and Pattern Recognition*, 0:413–420.

[Ramanan, 2011] Ramanan, D. (2011). Part-based models for finding people and estimating their pose. In Moeslund, T. B., Hilton, A., Krüger, V., and Sigal, L., editors, *Visual Analysis of Humans*, pages 199–223. Springer.

[Reiter and Mackworth, 1987] Reiter, R. and Mackworth, A. K. (1987). The logic of depiction. Technical report, Department of Computer Science, University of British Columbia, Vancouver, BC, Canada, Canada.

[Reiter and Mackworth, 1989] Reiter, R. and Mackworth, A. K. (1989). A logical framework for depiction and image interpretation. *Artif. Intell.*, 41(2):125–155.

[Rematas et al., 2012] Rematas, K., Fritz, M., and Tuytelaars, T. (2012). The pooled NBNN kernel: Beyond image-to-class and image-to-image. In Lee, K. M., Matsushita, Y., Rehg, J. M., and Hu, Z., editors, *Asian Conference on Computer Vision*, volume 7724 of *Lecture Notes in Computer Science*, pages 176–189. Springer.

[Roberts, 1963] Roberts, L. G. (1963). *Machine Perception of Three-Dimensional Solids*. Outstanding Dissertations in the Computer Sciences. Garland Publishing, New York.

[Rosenfeld, 1979] Rosenfeld, A. (1979). *Picture Languages: Formal Models for Picture Recognition*. Academic Press.

[Rosenfeld, 1982] Rosenfeld, A. (1982). Quadtree grammars for picture languages. *SMC*, 12:401–405.

[Russ et al., 1996] Russ, T., MacGregor, R., Salemi, B., Price, K., and Nevatia, R. (1996). Veil: Combining semantic knowledge with image understanding. In *ARPA Image Understanding Workshop*.

[Russ et al., 1998] Russ, T., University of Southern California Marina del Rey, Information Sciences Institute, Air Force Research Laboratory (Wright-Patterson Air Force Base, Ohio), Information Directorate, Rome Research Site, and United States Advanced Research Projects Agency (1998). *VEIL: Research in Knowledge Representation for Computer Vision*. AD-a341 150. Air Force Research Laboratory, Information Directorate, Rome Research Site.

[Russell et al., 2008] Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. (2008). LabelMe: A database and web-based tool for image annotation. *International Journal on Computer Vision*, 77(1-3):157–173.

[Rusu, 2009] Rusu, R. B. (2009). *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Computer Science Department, Technische Universitat Munchen, Germany.

[Rusu et al., 2009] Rusu, R. B., Blodow, N., and Beetz, M. (2009). Fast point feature histograms (FPFH) for 3D registration. In *IEEE International Conference on Robotics and Automation*, Kobe, Japan.

[Rusu et al., 2010] Rusu, R. B., Bradski, G., Thibaux, R., and Hsu, J. (2010). Fast 3d recognition and pose using the viewpoint feature histogram. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan.

[Rusu and Cousins, 2011] Rusu, R. B. and Cousins, S. (2011). 3D is here: Point cloud library (PCL). In *IEEE International Conference on Robotics and Automation*, Shanghai, China.

[Sadeghi and Farhadi, 2011] Sadeghi, M. and Farhadi, A. (2011). Recognition using visual phrases. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1745–1752.

[Sahbani et al., 2012] Sahbani, A., El-Khoury, S., and Bidaud, P. (2012). An overview of 3D object grasp synthesis algorithms. *Robot. Auton. Syst.*, 60(3):326–336.

[Sánchez and Latombe, 2003] Sánchez, G. and Latombe, J.-C. (2003). A single-query bi-directional probabilistic roadmap planner with lazy collision checking. In *Robotics Research*, volume 6 of *Springer Tracts in Advanced Robotics*, pages 403–417. Springer Berlin Heidelberg.

[Santos et al., 2006] Santos, P., Colton, S., and Magee, D. (2006). Predictive and descriptive approaches to learning game rules from vision data. In *Ibero-American Conference on Artificial Intelligence and Brazilian conference on Advances in Artificial Intelligence*, IBERAMIA-SBIA, pages 349–359, Berlin, Heidelberg. Springer-Verlag.

[Saxena et al., 2006] Saxena, A., Driemeyer, J., Kearns, J., and Ng, A. Y. (2006). Robotic grasping of novel objects. In *Advances in Neural Information Processing Systems*, pages 1209–1216.

[Saxena et al., 2008a] Saxena, A., Driemeyer, J., and Ng, A. Y. (2008a). Robotic grasping of novel objects using vision. *International Journal of Robotic Research*, 27(2):157–173.

[Saxena et al., 2008b] Saxena, A., Wong, L. L. S., and Ng, A. Y. (2008b). Learning grasp strategies with partial shape information. In *Conference on Artificial Intelligence*.

[Schmittwilken et al., 2009] Schmittwilken, J., Yang, M. Y., Förstner, W., and Plümer, L. (2009). Integration of conditional random fields and attribute grammars for range data interpretation of man-made objects. *Annals of GIS*, 15(2):117–126.

[Schölkopf and Smola, 2002] Schölkopf, B. and Smola, A. J. (2002). *Learning with kernels*. The MIT Press.

[Semeraro et al., 1996] Semeraro, G., Esposito, F., and Malerba, D. (1996). Ideal refinement of datalog programs. In Proietti, M., editor, *Logic Program Synthesis and Transformation*, volume 1048 of *Lecture Notes in Computer Science*, pages 120–136. Springer Berlin Heidelberg.

[Shanahan, 2005] Shanahan, M. (2005). Perception as abduction: Turning sensor data into meaningful representation. *Cognitive Science*, 29:103–134.

[Shapiro, 1983] Shapiro, L. G. (1983). Computer vision systems: Past, present, and future. In *Pictorial Data Analysis*, pages 199–235.

[Shapiro and Haralick, 1982] Shapiro, L. G. and Haralick, R. M. (1982). Organization of relational models for scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 4(6):595–602.

[Shawe-Taylor and Cristianini, 2004] Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, UK.

[Silberberg, 1987] Silberberg, T. M. (1987). Context dependent target recognition. In *Proc. of the Image Understanding Workshop*, pages 313–320, Los Angeles, CA.

[Siskind et al., 2007] Siskind, J. M., Sherman, J., Pollak, I., Harper, M. P., and Bouman, C. A. (2007). Spatial random tree grammars for modeling hierarchal structure in images with regions of arbitrary shape. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(9):1504–1519.

[Song et al., 2010] Song, D., Huebner, K., Kyrki, V., and Kragic, D. (2010). Learning task constraints for robot grasping using graphical models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1579 –1585.

[Sridhar et al., 2008] Sridhar, M., Cohn, A. G., and Hogg, D. C. (2008). Learning functional object-categories from a relational spatio-temporal representation. In *European Conference on Artificial Intelligence*, pages 606–610, Amsterdam, The Netherlands. IOS Press.

[Sridhar et al., 2010a] Sridhar, M., Cohn, A. G., and Hogg, D. C. (2010a). Relational graph mining for learning events from video. In *STAIRS: Starting AI Researchers' Symposium*, pages 315–327, Amsterdam, The Netherlands. IOS Press.

[Sridhar et al., 2010b] Sridhar, M., Cohn, A. G., and Hogg, D. C. (2010b). Unsupervised learning of event classes from video. In *Conference on Artificial Intelligence*. AAAI Press.

[Şucan et al., 2012] Şucan, A. I., Moll, M., and Kavraki, E. L. (2012). The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82.

[Sudderth et al., 2008] Sudderth, E. B., Torralba, A., Freeman, W. T., and Willsky, A. S. (2008). Describing visual scenes using transformed objects and parts. *International Journal on Computer Vision*, 77(1-3):291–330.

[Sweeney and Grupen, 2007] Sweeney, J. and Grupen, R. A. (2007). A model of shared grasp affordances from demonstration. In *Humanoids*, pages 27–35.

[Szeliski, 2010] Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer, New York.

[Teboul et al., 2013] Teboul, O., Kokkinos, I., Simon, L., Koutsourakis, P., and Paragios, N. (2013). Parsing facades with shape grammars and reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1744–1756.

[Tenenbaum et al., 1975] Tenenbaum, J., Barrow, H., Artificial Intelligence Center (SRI International), Stanford Research Institute Menlo Park California, United States Office of Naval Research, and Stanford Research Institute (1975). *MSYS: A System for Reasoning about Scenes*. Stanford research Institute. Artificial intelligence Center. Technical note. Stanford Research Institute.

[Tenenbaum and Barrow, 1977] Tenenbaum, J. M. and Barrow, H. G. (1977). Experiments in interpretation-guided segmentation. *Artificial Intelligence*, 8(3):241–274.

[Tenorth and Beetz, 2009] Tenorth, M. and Beetz, M. (2009). KnowRob – Knowledge processing for autonomous personal robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4261–4266.

[Terzic et al., 2010] Terzic, K., Hotz, L., and Sochman, J. (2010). Interpreting structures in man-made scenes - combining low-level and high-level structure sources. In *International Conference on Agents and Artificial Intelligence*, pages 357–364.

[Thon et al., 2008] Thon, I., Landwehr, N., and De Raedt, L. (2008). A simple model for sequences of relational state descriptions. In *European Conference on Machine Learning*, volume 5212, pages 506–521. Springer.

[Thrun and Wegbreit, 2005] Thrun, S. and Wegbreit, B. (2005). Shape from symmetry. In *International Conference on Computer Vision*, pages 1824–1831.

[Torralba et al., 2004] Torralba, A., Murphy, K. P., and Freeman, W. T. (2004). Sharing features: Efficient boosting procedures for multiclass object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 762–769.

[Toussaint et al., 2010] Toussaint, M., Plath, N., Lang, T., and Jetchev, N. (2010). Integrated motor control, planning, grasping and high-level reasoning in a blocks world using probabilistic inference. In *IEEE International Conference on Robotics and Automation*, pages 385–391.

[Tran and Davis, 2008] Tran, S. D. and Davis, L. S. (2008). Event modeling and recognition using Markov logic networks. In *European Conference on Computer Vision*, pages 610–623.

[Tuytelaars et al., 2011] Tuytelaars, T., Fritz, M., Saenko, K., and Darrell, T. (2011). The NBNN kernel. In *International Conference on Computer Vision*, pages 1824–1831.

[Tuytelaars and Mikolajczyk, 2007] Tuytelaars, T. and Mikolajczyk, K. (2007). Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280.

[Tylecek and Sara, 2011] Tylecek, R. and Sara, R. (2011). Modeling symmetries for stochastic structural recognition. In *IEEE International Conference on Computer Vision Workshops*, pages 632–639.

[Ullmann, 1983] Ullmann, J. (1983). Relational matching. In *Pictorial Data Analysis*, pages 147–170.

[Underwood and Coates, 1975] Underwood, S. and Coates, C. (1975). Visual learning from multiple views. *IEEE Transactions on Computers*, 24(6):651–661.

[Vapnik, 1995] Vapnik, V. N. (1995). *The nature of statistical learning theory.* Springer, New York.

[Vennekens et al., 2009] Vennekens, J., Denecker, M., and Bruynooghe, M. (2009). CP-logic: A language of causal probabilistic events and its relation to logic programming. *Theory and Practice of Logic Programming*, 9(3):245–308.

[Verbeke et al., 2012] Verbeke, M., Asch, V. V., Morante, R., Frasconi, P., Daelemans, W., and Raedt, L. D. (2012). A statistical relational learning approach to identifying evidence based medicine categories. In *EMNLP-CoNLL*, pages 579–589.

[Vogel and Schiele, 2007] Vogel, J. and Schiele, B. (2007). Semantic modeling of natural scenes for content-based image retrieval. *International Journal of Computer Vision*, 72(2):133–157.

[Waltz, 1975] Waltz, D. (1975). Understanding line drawings of scenes with shadows. In *The Psychology of Computer Vision*, page pages. McGraw-Hill.

[Wang and Fei-Fei, 2006] Wang, G. and Fei-Fei, Y. Z. L. (2006). Using dependent regions for object categorization in a generative framework. In *Conference on Computer Vision and Pattern Recognition*, pages 1597–1604. IEEE Computer Society.

[Winkler et al., 2012] Winkler, J., Bartels, G., Mösenlechner, L., and Beetz, M. (2012). Knowledge enabled high-level task abstraction and execution. *Conference for Advances in Cognitive Systems*, 2(1):131–148.

[Wu and Rehg, 2011] Wu, J. and Rehg, J. M. (2011). Centrist: A visual descriptor for scene categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1489–1501.

[Xu and Petrou, 2009] Xu, M. and Petrou, M. (2009). Learning logic rules for scene interpretation based on Markov logic networks. In *Asian Conference of Computer Vision*, pages III: 341–350.

[Yakimovsky and Feldman, 1973] Yakimovsky, Y. and Feldman, J. A. (1973). A semantics-based decision theory region analyzer. In *International Joint Conference on Artificial Intelligence*, pages 580–588, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Yang et al., 2008] Yang, L., Yang, J., Zheng, N., and Cheng, H. (2008). Layered object categorization. In *International Conference on Pattern Recognition*, pages 1–4.

[Yao and Fei-Fei, 2010] Yao, B. and Fei-Fei, L. (2010). Grouplet: A structured image representation for recognizing human and object interactions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9 –16.

[You and Fu, 1979] You, K. C. and Fu, K.-S. (1979). A syntactic approach to shape recognition using attributed grammars. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-9(6):334–345.

[Zhao et al., 2010] Zhao, P., Fang, T., Xiao, J., Zhang, H., Zhao, Q., and Quan, L. (2010). Rectilinear parsing of architecture in urban environment. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 342–349.

[Zhou et al., 2009] Zhou, X., Cui, N., Li, Z., Liang, F., and Huang, T. S. (2009). Hierarchical gaussianization for image classification. In *International Conference on Computer Vision*, pages 1971–1977.

[Zhu et al., 2010] Zhu, J., Li, L.-J., Li, F.-F., and Xing, E. P. (2010). Large margin learning of upstream scene understanding models. In *Advances in Neural Information Processing Systems*, pages 2586–2594.

[Zhu et al., 2012] Zhu, L., Chen, Y., Lin, Y., Lin, C., and Yuille, A. (2012). Recursive segmentation and recognition templates for image parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):359–371.

[Zhu et al., 2011] Zhu, L., Chen, Y., and Yuille, A. (2011). Recursive compositional models for vision: Description and review of recent work. *Journal of Mathematical Imaging and Vision*, 41(1-2):122–146.

[Zhu et al., 2007] Zhu, L., Chen, Y., and Yuille, A. L. (2007). Unsupervised learning of a probabilistic grammar for object detection and parsing. In *Advances in Neural Information Processing Systems 19*. MIT Press.

[Zhu et al., 2008] Zhu, L. L., Lin, C., Huang, H., Chen, Y., and Yuille, A. (2008). Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion. In *European Conference on Computer Vision*, pages 759–773, Berlin, Heidelberg. Springer-Verlag.

[Zhu and Mumford, 2006] Zhu, S.-C. and Mumford, D. (2006). A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision*, 2(4):259–362.

[Zhu and Yuille, 1996] Zhu, S. C. and Yuille, A. L. (1996). Forms: A flexible object recognition and modelling system. *International Journal of Computer Vision*, 20(3):187–212.

[Zhu and Ding, 2004] Zhu, X. and Ding, H. (2004). Planning force-closure grasps on 3D objects. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1258–1263.

Curriculum Vitae

Laura Antanas was born on July 6th 1981, in Timisoara, Romania. She finished high school at the "Grigore Moisil" Informatics High School in Timisoara. In 2001 she started studying computer engineering at the Polytechnic University of Timisoara, Automation and Computer Science Faculty. She received the degree of Master of Science from Ecole Nationale Supérieure des Télécommunications de Bretagne in 2007. In November 2007, she joined the DTAI (Declaratieve Talen en Artificiële Intelligentie) group of the Department of Computer Science to pursue a Ph.D. on (statistical) relation learning for computer vision and robotics under the supervision of prof. Luc De Raedt and prof. Tinne Tuytelaars. In 2012 she won the "Best Paper Award" at the International Conference on Pattern Recognition Applications and Methods for the paper *A Relational Distance-based Framework for Hierarchical Image Understanding*. In 2012 and 2013 her research was funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement 248258-First-MM.

Publication List

Journals

- Antanas, L., van Otterlo, M., Oramas M., J. A., Tuytelaars, T., and De Raedt, L. *There are plenty of places like home: Using relational representations in hierarchies for distance-based image understanding.* Neurocomputing Journal, volume 123, pages 75-85, 2014.
- Janssens, T., Antanas, L., Derde, S., Vanhorebeek, I., Van den Berghe, G., Guiza Grandas, F. *Charisma: An integrated approach to automatic H&E-stained skeletal muscle cell segmentation using supervised learning and novel robust clump splitting.* Medical Image Analysis, volume 17, issue 8, pages 1206-1219, 2013.

Conferences and Workshops

- Antanas, L., Hoffmann, M., Frasconi, P., Tuytelaars, T., and De Raedt, L. *A relational kernel-based approach to scene classification.* In Proceedings of Workshop on Applications of Computer Vision, pages 133-139, 2013.
- Neumann, M., Moreno, P., Antanas, L., Garnett, R., Kersting, K. *Graph kernels for object category prediction in task-dependent robot grasping.* In Online Proceedings of the Eleventh Workshop on Mining and Learning with Graphs, pages 1-6, 2013.
- Billiet, L., Oramas M., J., Hoffmann, M., Meert, W., Antanas, L. *Rule-based hand posture recognition using qualitative finger configurations acquired with the Kinect.* In Proceedings of the 2nd International Conference on Pattern Recognition - Applications and Methods, pages 539-542, 2013.

- Moldovan, B., Antanas, L., Hoffmann, M. *Opening doors: An initial SRL approach.* In Lecture Notes in Computer Science Post Proceedings, Inductive Logic Programming, Springer, pages 178-192, 2013.
- Robben, D., Smeets, D., Ruijters, D., Hoffmann, M., Antanas, L., Maes, F., Suetens, P. *Intra-patient non-rigid registration of 3D vascular cerebral images.* In Lecture Notes in Computer Science, MICCAI Workshop on Clinical Image-based Procedures: From Planning to Intervention, Springer, pages 106-113, 2013.
- Antanas, L., van Otterlo, M., Oramas Mogrovejo, J. A., Tuytelaars, T., and De Raedt, L. *A relational distance-based framework for hierarchical image understanding.* In Proceedings of the 1st International Conference on Pattern Recognition - Applications and Methods, pages 206-218, 2012, *Best Paper Award.*
- Antanas, L., Frasconi, P., Costa, F., Tuytelaars, T., and De Raedt, L. *A relational kernel-based framework for hierarchical image understanding.* In Lecture Notes in Computer Science, Structural, Syntactic, and Statistical Pattern Recognition, Springer, pages 171-180, 2012.
- Derde, M., Antanas, L., De Raedt, L., Guiza Grandas, F. *An interactive learning approach to histology image segmentation.* In Proceedings of the 24th Benelux Conference on Artificial Intelligence, pages 1-8, 2012.
- Janssens, T., Antanas, L., Derde, S., Vanhorebeek, I., Van den Berghe, G., Guiza Grandas, F. *Charisma: An Integrated Approach to Automatic H&E-stained Skeletal Muscle Cell Segmentation Using Supervised Learning and Novel Robust Clump Splitting Techniques.* In Bioimaging, abstract, 2012.
- Antanas, L., Frasconi, P., Tuytelaars, T., and De Raedt, L. *Employing logical languages for image understanding.* In IEEE Workshop on Kernels and Distances for Computer Vision, International Conference on Computer Vision, pages 1-2, 2011.
- Antanas, L., van Otterlo, M., Oramas Mogrovejo, J. A., Tuytelaars, T., and De Raedt, L. *Not far away from home: A relational distance-based approach to understand images of houses.* In Lecture Notes in Computer Science, Inductive Logic Programming, Springer, pages 22-29, 2010.
- Antanas, L., Gutmann, B., Thon, I., Kersting, K., and De Raedt, L. *Combining video and sequential statistical relational techniques to monitor card games.* In Proceedings of the ICML Workshop on Machine Learning and Games, pages 1-6, 2010.

- Antanas, L., Gutmann, B., Thon, I., Kersting, K., and De Raedt, L. *Combining video and sequential statistical relational techniques to monitor card games.* In Proceedings of the Belgian-Dutch Conference on Machine Learning, pages 1-6, 2010.
- Antanas, L., Thon, I., van Otterlo, M., Landwehr, N., and De Raedt, L. *Probabilistic logical sequence learning for video.* Online Proceedings In Inductive Logic Programming, pages 1-6, 2009.
- Antanas, L., van Otterlo, M., De Raedt, L., and Thon, I. *Learning probabilistic relational models from sequential video data with applications in table-top and card games.* In Proceedings of the Belgian- Dutch Conference on Machine Learning, pages 1-2, 2009.
- Antanas, L., Driessens, K., Croonenborghs, T., Ramon, J. *Using decision trees as the answer network in temporal-difference networks.* In Proceedings of the 18th European Conference on Artificial Intelligence, pages 847-848, 2008.

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
DECLARATIVE LANGUAGES AND ARTIFICIAL INTELLIGENCE
Celestijnenlaan 200A 3001 Heverlee
B-3001 Heverlee
first.name@dept.kuleuven.be
www.website.kuleuven.be